DOCTEUR EN SCIENCES

Développement d'une base de données relationnelle permettant de filtrer des groupes de gènes biologiquement pertinents dans l'analyse des données de damiers à ADN

Bareke, Eric

Award date:
2011

Awarding institution:
Universite de Namur

Facultés Universitaires Notre Dame de la Paix

Faculté de Sciences

Département de Biologie

Development of a Relational Database that allow Filtering of Biologically Relevant Groups of Genes in Microarray Data Statistical Analysis.

Dissertation présentée par
Eric Bareke
en vue de l'obtention du grade
de **Docteur en Sciences**

Composition du jury :

Prof. Eric Depiereux (**Promoteur**, secrétaire du jury, Département de Biologie, FUNDP)
Prof. Naji Habra (**Co-promoteur**, membre du jury, Faculté d'Informatique, FUNDP)
Prof. Jean Jacques Lettesson (Président du jury, Département de Biologie, FUNDP)
Prof. Marcel Rémon (Membre du jury, Département de Mathématique, FUNDP)
Prof. Xavier Debolle (Membre du jury, Département de Biologie, FUNDP)
Dr. Christophe Lambert (Membre du jury, PROGENUS S.A., Gembloux)

2011

Acknowledgements

I would like to address my warmest appreciation to my thesis director Prof. Eric Depiereux for his guidance and encouragement during my years as a Ph.D. student. He has been more than a thesis director and without him I would not have come to Bioinformatics and will always be grateful for all he did for me. Without his inspiration, this thesis would not exist.

My thanks also go to my thesis co-director Prof. Naji Habra for his help and wise support during the course of this degree. His technical advices and willingness to help allowed me to strengthen my computing skills.

I would also like to thank Dr. Fabrice Berger, Benoit De Hertogh, Michael Pierre, Bertrand De Meulder and Sophie Depiereux for many years of nice collaborations. This thesis wouldn't look the same without you!

Other persons that have affected my research in a positive way and deserve my thanks are Prof. Marcel Rémon, Prof. Xavier Debolle and Christophe Lambert.

Furthermore, I would like to thank all my Masters Degree level lecturers for having introduced me to several areas of bioinformatics.

I am also grateful to the Kingdom of Belgium (through the Belgian Technical Cooperation – CTB-BTC) and the University of Namur (Facultés Universitaires Notre Dame de la Paix – FUNDP) for funding this work.

A special appreciation goes to Dr. Desire Ndushabandi and Révérien Rutayisire who cared about my family during my absence.

Finally, I would like to thank my family members who always encouraged and supported me all this time.

I cannot conclude my acknowledgements, without mentioning my wife Jacqueline (Lynn) Mukarusagara who stood by me all these years, showing endless love and displaying patience and understanding. For being the light of my life and the dearest friend imaginable, I would like to conclude by extending my most cordial and endless thanks to her.

Thank you all!

Eric Bareke

b

Abstract

With the advent of microarray technology, new approaches have led to promising discoveries in life sciences. Microarrays have become a useful tool in biomedical research as they can be used to determine simultaneously the relative expression of thousands of genes in a given sample.

However, the impressive quantity of data microarrays provide is associated with a substantial amount of noise. Microarray gene expression data are commonly perceived as being extremely noisy because of many imperfections inherent in the current technology. If the noise is consistent and reproducible it can be filtered from the data and some false positives can be eliminated.

There are two principal sources of noise in microarray experiments: biological noise and technical noise.

Biological noise consists of variation among conditions. Technical noise consists of differences in sample preparation and experiment variables which include nonspecific cross hybridization, differences in the efficiency of labeling reactions and production differences between microarrays. Biological noise cannot be corrected but it can be accounted for with statistics using replicates of the treatments or conditions.

However, it is widely believed that a high level of technical noise in microarray data is the most critical deterrent to the successful use of this technology in studies of normal and abnormal biological processes. In particular, the notorious lack of reproducibility of lists of detected genes across platforms and laboratories, as well as validation problems associated with prognostic signatures, is frequently attributed to this "flaw" of microarray technology.

Although several recently proposed analysis methods for microarray data can cope with heavy tailed noise, many of them rely on statistical assumptions. These methods however can only reduce noise.

This thesis contributes with two papers that pursue the novel approach of reducing microarray data noise. We present a novel concept to organize microarray data to ease finding of new pathways and validation of existing ones by using refined microarray meta-data and a novel co-expression networks visualization tools with advanced features to manipulate generated (sub) networks.

These approaches consist of applying complex computational processes to enrich microarray meta-data annotations by use of different biological information.

The resulting applications have been carefully evaluated and confirm their biological phenomenon’s predictive and validation powers.

d

Contents

g

a

Introduction

Background

Microarrays, a new technology that makes it possible to examine the expression of thousands of genes simultaneously, opened up an entire new world to researchers (CAMPBELL, et al., 2011) .They are now not restricted to studying a unique single aspect of a life mechanism; instead one can explore a genome-wide view of complex interactions (Wu, et al.). This new technology is helping scientists discover and understand disease pathways, and ultimately develop better methods of detection, treatment, and prevention of different diseases (Drews, 2000). Data generated from whole-genome microarray studies are richer and deeper than ever before.

Data from a single array experiment – whether gene expression or DNA analysis – can often be used for a number of different studies that otherwise would have required the compilation of data from numerous independent experiments. For example, the same expression data from a study could be used to understand the mechanism of another study. Moreover, arrays are being designed to simultaneously monitor whole-genome, providing a complete view of how different biological mechanisms operate. Also, the flexibility of microarray analysis allows a single array and a single experiment to encompass different types of studies.

Overview of microarrays

The final deciphering of the complete human genome, together with the improvement of high throughput technologies such as Microarrays (Pease, et al., 1994) (Schena, et al., 1995), is causing a fundamental transformation in life sciences research. This technique makes it possible to examine the expression of thousands of genes simultaneously (Shalon, et al., 1996).

By using a microarray containing many DNA samples, researchers can determine, in a single experiment, the expression levels of hundreds or thousands of genes within a cell by measuring the amount of mRNA bound to each site on the array (Shalon, et al., 1996). With the aid of a computer, the amount of mRNA bound to the spots on the microarray is precisely measured, generating a profile of gene expression in the cell.

Gene expression

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product (Figure 1). These products are often proteins, but in non-protein coding genes such as ribosomal RNA (rRNA) genes or transfer RNA (tRNA) genes, the product is a functional RNA. The process of gene expression is used by all known life - eukaryotes (including multicellular organisms), prokaryotes (bacteria and archaea) and viruses - to generate the macromolecular machinery for life. Several steps in the gene expression process may be modulated, including the transcription, RNA splicing, translation, and post-translational modification of a protein.

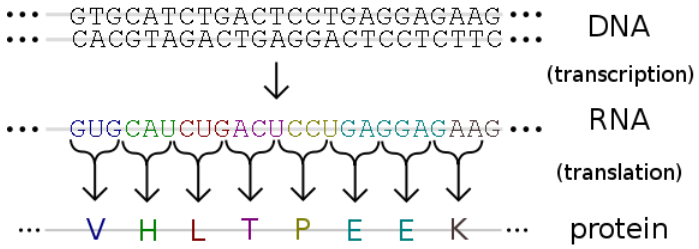


Figure 1 - Molecular biology paradigm (Source: http://upload.wikimedia.org/Genetic_code.png)

What is a Microarray?

A Microarray is a new powerful tool for studying the molecular basis of interactions on a scale that is impossible using conventional analysis (Wilson, et al., 2006). It consists of an arrayed series of thousands of microscopic spots of DNA oligonucleotides, called features. Each of these features contains picomoles (10^{-12} moles) of a specific DNA sequence, known as probes (or reporters) (Huynh, et al., 2009) (Ganguly, et al., 2010). These probes can be a short section of a gene or other DNA elements that are used to hybridize a cDNA (Pollack, et al., 1999) or cRNA sample (called target) under high-stringency conditions (Figure 2). Probe-target hybridization is usually detected and quantified by detection of fluorophore-, silver-, or chemiluminescence-labeled targets to determine relative abundance of nucleic acid sequences in the target (Shalon, et al., 1996). Since an array can contain tens of thousands of probes, a microarray experiment can accomplish many genetic tests in parallel. Therefore arrays have dramatically accelerated many types of investigation (Bell, et al., 2010).

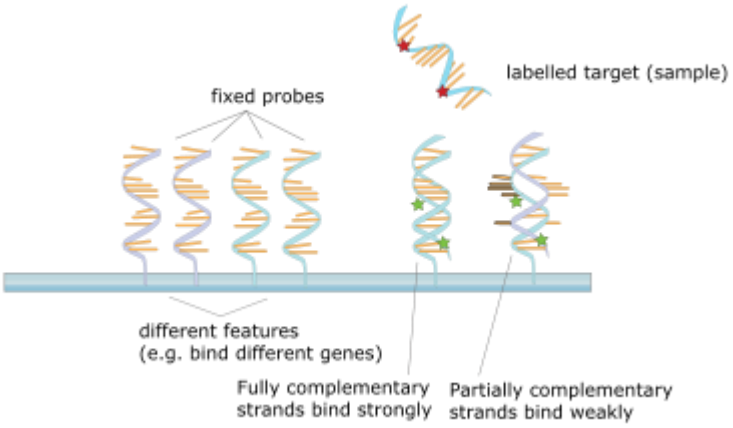


Figure 2 - Hybridization principle (Source: http://upload.wikimedia.org/NA_hybrid.png)

In standard microarrays, the probes are attached via surface engineering to a solid surface by a covalent bond to a chemical matrix (via epoxy-silane, amino-silane, lysine, polyacrylamide or others) (Figure 2). The solid surface can be glass or a silicon chip, in which case they are colloquially known as an Affychip[1] or GeneChip when an Affymetrix chip is used. Other microarray platforms, such as Illumina[2], use microscopic beads, instead of the large solid support. DNA arrays are different from other types of microarray only in that they either measure DNA (Pollack, et al., 1999) or use DNA as part of its detection system (Shalon, et al., 1996).

The core principle behind microarrays is hybridization between two DNA strands, the property of complementary nucleic acid sequences to specifically pair with each other by forming hydrogen bonds between complementary nucleotide base pairs. A high number of complementary base pairs in a nucleotide sequence mean tighter non-covalent bonding between the two strands. After washing off of non-specific bonding sequences, only strongly paired strands will remain hybridized. So fluorescently labeled target sequences that bind to a probe sequence, generate a signal that depends on the strength of the hybridization. The strength of hybridization, however, is determined by the number of paired bases, the hybridization conditions and washing after hybridization. Total strength of the signal, from a spot depends upon the amount of target sample binding to the probes present on that spot (Fang).

In addition, microarrays use relative quantization in which the intensity of a feature is compared to the intensity of the same feature under a different condition (Sassolas,

[1] www.affymetrix.com

[2] www.illumina.com

et al., 2007), and the identity of the feature is known by its position on the array (Figure 3).

Many types of arrays exist and the broadest distinction is whether they are spatially arranged on a surface or on coded beads:

- The traditional solid-phase array is a collection of orderly microscopic "spots", called features, each with a specific probe attached to a solid surface, such as glass, plastic or silicon biochip (commonly known as a genome chip, DNA chip or gene array). Thousands of them can be placed in known locations on a single DNA microarray.
- The alternative bead array is a collection of microscopic polystyrene beads, each with a specific probe and a ratio of two or more dyes, which do not interfere with the fluorescent dyes used on the target sequence.

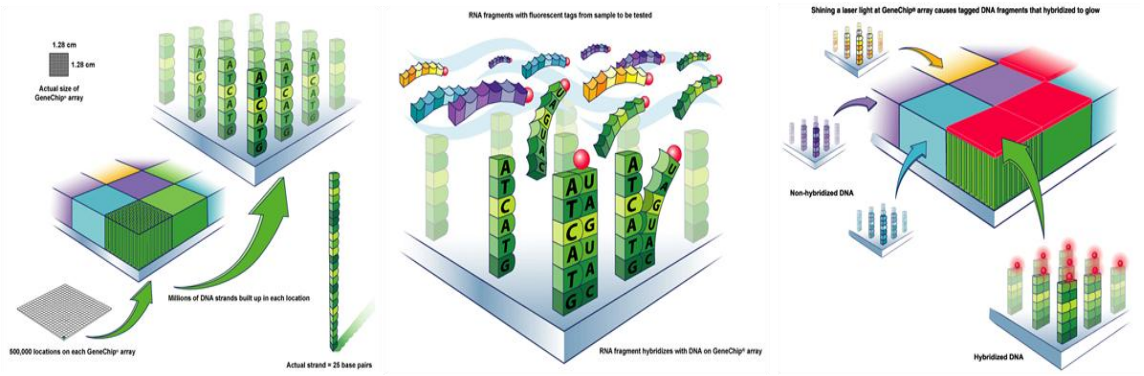


Figure 3 - Example of an Affymetrix Genechip manufacture process and working principle (Sources: http://www.affymetrix.com/GeneChip.gif)

Categories of DNA microarrays

Microarrays can be categorized in different ways, depending on the number of probes under examination, costs, customization requirements, number of channels (dyes) and the type of scientific question being asked. Arrays may have as few as 10 probes or up to 2.1 million micrometer-scale probes from commercial vendors (Kumar, 2011).

Categorization by fabrication technologies

Microarrays can be fabricated using a variety of technologies, including printing with fine-pointed pins onto glass slides, photolithography using pre-made masks, photolithography using dynamic micro mirror devices, ink-jet printing, or electrochemistry on microelectrode arrays (Kumar, 2011) (XiaoKun and HuaSheng, 2009).

Spotted microarrays

In **spotted microarrays**, the probes are cDNA or small fragments of PCR[3] products that correspond to mRNAs (Hubank and Schatz, 1994). The probes are synthesized prior to deposition on the array surface and are then "spotted" onto glass. A common approach utilizes an array of fine pins or needles controlled by a robotic arm that is dipped into wells containing DNA probes and then depositing each probe at designated locations on the array surface (Schena, et al., 1995) (Eisen and Brown, 1999). The resulting "grid" of probes represents the nucleic acid profiles of the prepared probes and is ready to receive complementary cDNA or cRNA "targets" derived from experimental or clinical samples.

This technique is used by research scientists around the world to produce "in-house" printed microarrays from their own labs. These arrays may be easily customized for each experiment, because researchers can choose the probes and printing locations on the arrays, synthesize the probes in their own lab (or collaborating facility), and spot the arrays. They can then generate their own labeled samples for hybridization, hybridize the samples to the array, and finally scan the arrays with their own equipment. This approach provides a relatively low-cost microarray that may be customized for each study, and avoids the costs of purchasing often more expensive commercial arrays that may represent vast numbers of genes that are not of interest to the investigator.

[3] PCR refers to Polymerase Chain Reaction

oligonucleotide microarrays

In **oligonucleotide microarrays**, the probes are short sequences designed to match parts of the sequence of known or predicted open reading frames. Although oligonucleotide probes are often used in "spotted" microarrays, the term "oligonucleotide array" most often refers to a specific technique of manufacturing.

Oligonucleotide arrays are produced by printing short oligonucleotide sequences designed to represent a single gene or family of gene splice-variants (Maskos and Southern, 1992) (Fodor, et al., 1993). This is achieved by synthesizing this sequence directly onto the array surface instead of depositing intact sequences (Schena, et al., 1995). Sequences may be longer (60-mer probes such as the Agilent design) or shorter (25-mer probes [4]produced by Affymetrix) depending on the desired purpose. Longer probes are more specific to individual target genes while shorter probes may be spotted in higher density across the array and are cheaper to manufacture (Eisen and Brown, 1999).

One technique used to produce oligonucleotide arrays include photolithographic synthesis (Agilent[5] and Affymetrix (Figure 4)) on a silica substrate where light and light-sensitive masking agents are used to "build" a sequence one nucleotide at a time across the entire array (Pease, et al., 1994). Each applicable probe is selectively "unmasked" prior to bathing the array in a solution of a single nucleotide, then a masking reaction takes place and the next set of probes are unmasked in preparation for a different nucleotide exposure. After many repetitions, the sequences of every probe become fully constructed. More recently, Maskless Array[6] Synthesis from NimbleGen Systems has combined flexibility with large numbers of probes (Nuwaysir, et al., 2002).

[4] This is a conventional length for oligonucleotide probe.

[5] www.agilent.com

[6] www.nimblegen.com

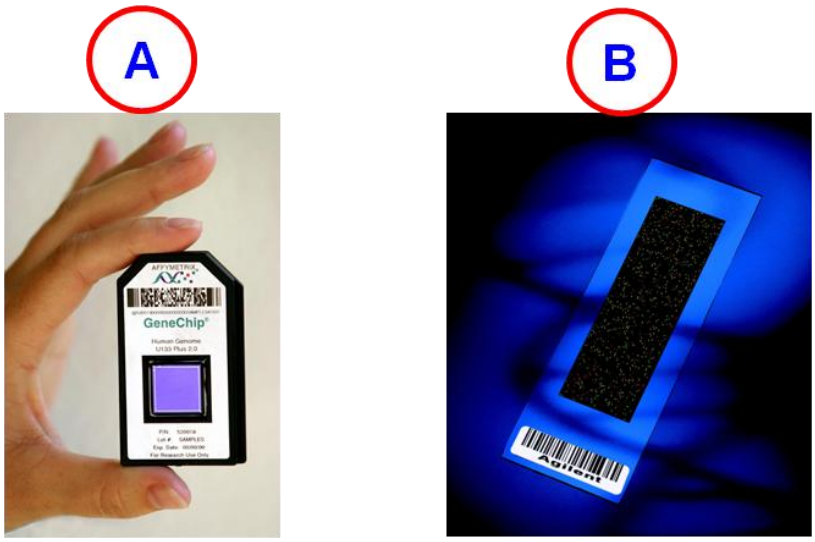


Figure 4 - Examples of oligonucleotide microarrays: AFFYMETRIX & AGILENT

(A; source: http://www.affymetrix.com) – Agilent (B; source: http://www.agilent.com)

Categorization based on number of dyes

Considering channels (dyes) number criteria, microarrays can be subdivided into two categories:

Two-dye microarrays

Two-color microarrays or two-channel microarrays are typically hybridized with cDNA prepared from two samples to be compared (e.g. diseased tissue versus healthy tissue) and that are labeled with two different fluorophores (Schena, et al., 1995). Fluorescent dyes commonly used for cDNA labeling include Cy3, which has a fluorescence emission wavelength of 570 nm (corresponding to the green part of the light spectrum), and Cy5 with a fluorescence emission wavelength of 670 nm (corresponding to the red part of the light spectrum) (Welford, et al., 1998). The two Cy-labeled cDNA samples are mixed and hybridized to a single microarray that is then scanned in a microarray scanner to visualize fluorescence of the two fluorophores after excitation with a laser beam of a defined wavelength (Widengren and Schwille, 2000) (Huang, et al., 2005). Relative intensities of each fluorophore may then be used to identify up-regulated and down-regulated genes (Tang, et al., 2007).

One-dye microarrays

In single-channel microarrays or one-color microarrays, the arrays provide intensity data for each probe or probe set indicating a relative level of hybridization with the labeled target (Torimura, et al., 2001) (Heinlein, et al., 2003). However, they do not truly indicate abundance levels of a gene but rather relative abundance when compared to other samples or conditions when processed in the same experiment. Each RNA molecule encounters protocol and batch-specific bias during amplification, labeling, and hybridization phases of the experiment making comparisons between genes for the same microarray uninformative (Widengren and Schwille, 2000) (Huang, et al., 2005).

The comparison of two conditions for the same gene requires two separate single-dye hybridizations. Several popular single-channel systems are the Affymetrix "Gene Chip", Illumina "Bead Chip", and Agilent single-channel arrays. One strength of the single-dye system lies in the fact that an aberrant sample cannot affect the raw data derived from other samples, because each array chip is exposed to only one sample (Harvey and Levitus, 2009) (as opposed to a two-color system in which a single low-quality sample may drastically impinge on overall data precision even if the other sample was of high quality). Another benefit is that data are more easily compared to arrays from different experiments so long as batch effects have been accounted for (Chen, et al., 2009). A drawback to the one-color system is that, when compared to the two-color system, twice as many microarrays are needed to compare samples within an experiment.

Experimental design

Due to the biological complexity of gene expression, the considerations of experimental design are of critical importance if statistically and biologically valid conclusions are to be drawn from DNA microarray data.

There are three main elements to consider when designing a microarray experiment.

First, replication of the biological samples is essential for drawing conclusions from the experiment.

Second, technical replicates (two RNA samples obtained from each experimental unit) help to ensure precision (Cui, et al., 2005) and allow for testing differences within treatment groups.

The biological replicates include independent RNA extractions while technical replicates may be two aliquots of the same extraction.

Third, spots of each cDNA clone or oligonucleotide are present as replicates (at least duplicates) on the microarray slide, to provide a measure of technical precision for each hybridization.

Nevertheless, while it is preferable to have as many replicates as possible (Lin, et al., 2002), the high cost of chips makes it more practical; hence the use of smaller number of replicates is preferred. To determine the number of replicates, biologists must balance confidence, cost and efficiency against the desire to explore more experimental conditions.

Regardless of the replication strategy, sample-to-sample variability cannot, in general, be completely minimized (Nagele, 2003). Thus, it is important to consider the best ways to maximize consistency between samples. In many types of studies, it is not possible to control completely all variables, and there may be considerable variability due to experimental difficulties or individual genetic variation. But such factors do not preclude the discovery of some genes that are consistently different and that clearly ‘cluster’ or differentiate between the sample sets.

Standardization

Microarray data is difficult to exchange due to the lack of standardization in platform fabrication, assay protocols, and analysis methods (Mwololo, et al., 2010). This presents an interoperability problem in bioinformatics. Various grass-roots open-source projects are trying to ease the exchange and analysis of data produced with non-proprietary chips:

For example, the "Minimum Information About a Microarray Experiment" (MIAME) (Brazma, et al., 2001) checklist helps define the level of detail that should exist. However MIAME does not describe the format for the information (no verification of complete semantic compliance).

Statistical Data analysis

Analytical precision is influenced by a number of variables. Challenges include taking into account effects of background noise and appropriate normalization of the data (Bolstad, 2002). Different normalization methods (Figure 5) may be suited to specific platforms and, in the case of commercial platforms, the analysis may be proprietary.

Some algorithms that affect statistical analysis include:

- **Image analysis:** gridding, spot recognition of the scanned image (segmentation algorithm), removal or marking of poor-quality and low-intensity features (called flagging) (Korn, et al., 2004).

- **Data processing:** background subtraction (based on global or local background), determination of spot intensities and intensity ratios, visualization of data (e.g. see MA plot), and log-transformation of ratios, global or local normalization of intensity ratios (Wang, et al., 2001).

- **Identification of statistically significant changes:** t-test, ANOVA, Bayesian method (Ben-Gal, et al., 2005) Mann–Whitney test methods tailored to microarray data sets, which take into account multiple comparisons (Leung and Cavalieri, 2003) or cluster analysis (Priness, et al., 2007). These methods assess statistical power based on the variation present in the data and the number of experimental replicates, and can help minimize Type I and type II errors in the analyses.

- **Network-based methods:** Statistical methods that take the underlying structure of gene networks into account, representing either associative or causative interactions or dependencies among gene products (Wei, et al., 2004).

Sometimes, microarray data may require further processing aimed at reducing the dimensionality of the data to aid comprehension and more focused analysis (Wouters, et al., 2003). Other methods permit analysis of data consisting of a low number of biological or technical replicates.

For example, the Local Pooled Error (LPE) test pools standard deviations of genes with similar expression levels in an effort to compensate for insufficient replication (Jain, et al., 2003).

For the purpose of this research, in this thesis, we limit our analytical discussion on Affymetrix-made microarrays while there are an optimal number of biological or technical replicates.

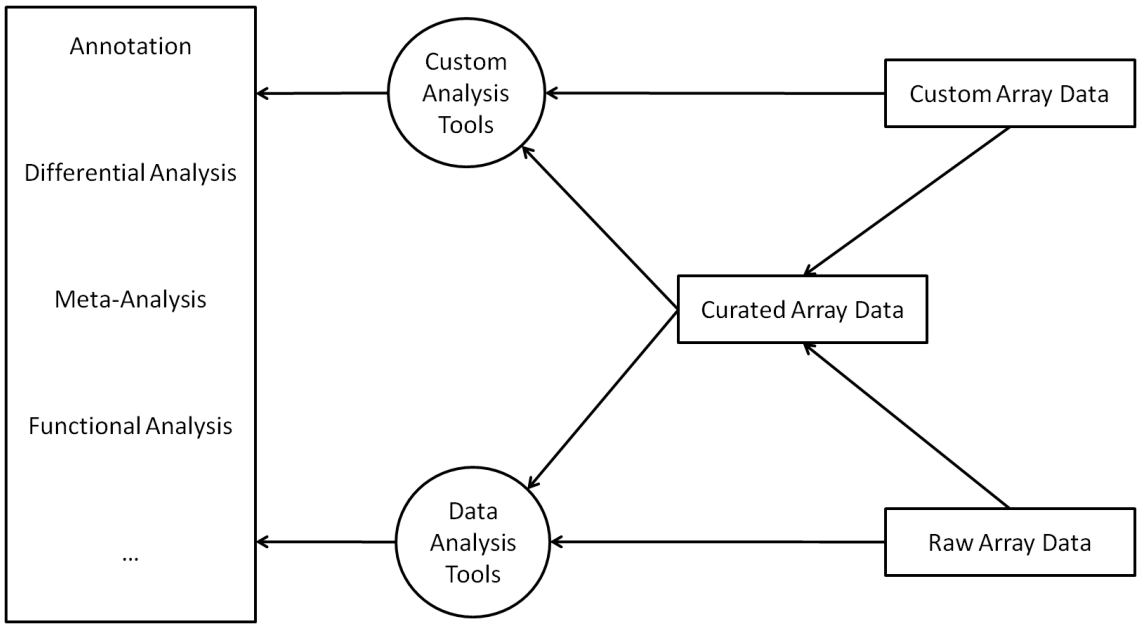


Figure 5 - Example of steps involved in most of microarray data analysis

The major issue when studying datasets with few replicates is the decreased power of the analysis (Shriner and Vaughan). To compensate for this lack of information about expression data generated by microarrays, some researchers have tried to work at the probe level, breaking down probesets into small components called probes. In Affymetrix technology, each gene is typically represented by a set of 11-20 pairs of probes. In order to obtain expression measures, probe level data are summarized by application of a series of improved preprocessing methods (Yin, et al.).

Probe-level Analysis

There are dozens (thousands if you consider mixing and matching) of approaches to preprocessing (Nadon and Shoemaker, 2002) but they are generally classified into the following steps:

- Background adjustment (this step combine background correction PM correction in many methods)
- Normalization
- Summarization

Professor Eric Depiereux bioinformatics team have put extraordinary efforts on developing new probe-level statistical methods [7]ranging from data filters to novel analysis approaches. Several such methods have been developed so far (Goeman and Buhlmann, 2007), but, this research focuses on some popular (Table 1) methods which comprise basic tools for much experimental work.

Due to the fact that the Professor Depiereux bioinformatics team focuses only on Affymetrix platform, our discussion in this thesis is limited to Affymetrix DNA microarrays, other existing platforms falling beyond the scope of this thesis work.

[7] My DEA thesis "Analyse statistique des micropuces à ADN: Développement d'un filtre sur les probes par le biais de matrices de fréquences visant à élimininer les probes non-performants" focused mainly on filtering noise at probe-level. I developped a frequency table correlation based filter to remove non-performing probes).

Table 1 - Summary of major Affymetrix microarray data preprocessing methods

Methods	Background correction	Normalization	Perfect Match correction	Summarization
RMA (Robust Multichip Averaging)	Rma (Bolstad, et al., 2003)	quantiles	pmonly	medianpolish
MAS (Microarray Affymetrix Suite)	Mas (Affymetrix, 2001)	mas	mas	mas
GCRMA	Gcrma (Wu, et al., 2004)	quantiles	pmonly	medianpolish
Custom Methods	rma rma2 mas gcrma-eb gcrma-mle	quantiles quantiles.robust loess contrast constant invariantset qspline vsn	pmonly mas subtractmm	medianpolish avdiff liwong mas playerout rlm

Comparison of popular Affymetrix GeneChip pre-processing methods and the way some of them are combined as a single methodology into major R packages. These packages are (some of them) proprietary packages and license issues apply.

Major Affymetrix-related preprocessing steps

Background adjustment

The first step is generally to **background adjustment** the intensity reading for each spot. Background fluorescence can arise from many sources, such as non-specific binding of labeled sample to the array surface, processing effects such as deposits left after the wash stage or optical noise from the scanner (Wu, et al., 2004). There is always some level of background noise, even when nothing but sterile water is labeled and hybridized to the array, some fluorescence will still be picked up by the scanner. Different algorithms will use different methods of background correction. Only three background adjustment methods will be described here.

RMA convolution

This is an implementation of the background adjustment carried out as part of the RMA method (Irizarry, et al., 2003). These authors found various problems with using the Mismatch (MM) probes in the preprocessing steps and proposed a procedure that uses only the Perfect Match (PM) intensities. In this procedure, the PM values are corrected, array by array, using a global model for the distribution of probe intensities. The model was motivated by the empirical distribution of probe intensities. In particular the observed PM probes are modeled as the sum of a Gaussian noise component, ***N***, with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma^2}$ and an exponential signal component, ***S***, with mean $\boldsymbol{\alpha}$. To avoid the possibility of negatives expression values, the normal distribution is truncated at zero. Given that we have the observed intensity, ***X***, then this leads to the following adjustment.

$$
\begin{aligned}
&S = X + N, \\
&where \sim \mathrm{N} \sim N(\mu, \sigma^2) \; and \; X \sim \mathrm{Exp}(\alpha) \\
&X / S \sim \mathrm{N(a, b, 0, S)} \\
&\mathrm{a} = \mathrm{S} - \mu - \sigma^2 \alpha \\
&b = \sigma \\
&E(X / S) = a + b \frac{\phi(\frac{a}{b}) - \phi(\frac{S-a}{b})}{\Phi(\frac{S-a}{b}) + \Phi(\frac{a}{b}) - 1} \\
&V(X / S) = b^2 (1 - \frac{\frac{a}{b}\phi(\frac{a}{b}) - \frac{S-a}{b}\phi(\frac{S-a}{b})}{\Phi(\frac{S-a}{b}) + \Phi(\frac{a}{b}) - 1} - (\frac{\phi(\frac{a}{b}) - \phi(\frac{S-a}{b})}{\Phi(\frac{S-a}{b}) + \Phi(\frac{a}{b}) - 1})^2) \\
&g(S) = E(X / S) \\
&h(S) = V(X / S) \\
&(S_1, \ldots, S_n) being \; a \; sample \; of \; obersved \; \mathrm{int} ensities \; and \\
&(g(S_1), \ldots, g(S_n)) \; being \; background - corrected \; values.
\end{aligned}
$$

Equation 1 - RMA convolution method computation

MAS 5.0 background correction

In this background adjustment method (Affymetrix, 2002), the chip is divided into a grid of k (default k = 16) rectangular regions. For each region, the lowest 2% of probe intensities are used to compute a background value for that grid. Then probe intensity is adjusted based upon a weighted average of each of the background values. The weights are dependent on the distance between the probe and the centroid of the grid. In particular, the weights are:

$$\text{For a single probe with coordinates } (a,b):$$
$$d_k(a,b) = \text{distance to center of zone } k$$
$$w_k(a,b) = \frac{1}{d_k^2(a,b) + s_0}\ ,\ w_k(a,b) \text{ being the weight and } s_0 \text{ the smoothing value.}$$
$$\text{Default } s_0 = 100$$
$$\text{Background for probe at } (a,b):$$
$$l(a,b) = \left[\sum_k w_k(a,b)\ b\ Z_k\right] x \left[\sum_k w_k(a,b)\right]^{-1}$$
$$\text{Local noise for probe at } (a,b):$$
$$n(a,b) = \left[\sum_k w_k(a,b)\ n\ Z_k\right] x \left[\sum_k w_k(a,b)\right]^{-1}$$
$$\text{For cell intensity } I(a,b): I'(a,b) = \max\{I(a,b),\ \text{Fraction of global background variation (default } f = 0.5)\}$$

$$\text{Background} - \text{adjusted intensity}:$$
$$A(a,b) = \max\{I(a,b) - l(a,b),\ f.n(a,b))\}$$

Equation 2 - MAS 5.0 background adjustment computation

Special care is taken to avoid negative values or other numerical problems for low intensity regions. Note this method corrects both PM and MM probes.

Ideal mismatch

Originally, the suggested purpose of the MM probes was that they could be used to adjust the PM probes (Affymetrix, 2001) for probe-specific non-specific binding by subtracting the intensity of the MM probe from the intensity of the corresponding PM probes. However, this becomes problematic because, for data from a typical array, as many as 30% of MM probes have intensities higher than their corresponding PM probe (Naef and Magnasco, 2003).

Thus, when raw MM intensities are subtracted from the PM intensities many negative expression values result, which makes little sense, because an expression value should not be below zero.

Another drawback is that the negative values preclude the use of logarithms. To remedy the negative impact of using raw MM values, Affymetrix introduced the concept of an **Ideal Mismatch (IM)** which was guaranteed, by design, to be smaller than the corresponding PM intensity. The goal is to use MM when it is physically possible and a quantity smaller than the PM in other cases. This is done by computing the specific background, (***SB***), for each probeset. This is a robust average of the log ratios of PM to MM for each probe pair in the probeset. If ***i*** is the probe and ***k*** is the probeset, then for the probe pair indexed by ***i*** and **k** the ideal mismatch IM is given by:

$$IM_{k,i} = \begin{cases} MM_{k,i} & MM_{k,i} < PM_{k,i} \\ \dfrac{PM_{k,i}}{2^{SB_k}} & MM_{k,i} \geq PM_{k,i} \quad and \quad SB_k > \tau_c \\ \dfrac{PM_{k,i}}{2^{\frac{\tau_c}{(1+\frac{\tau_c - SB_k}{\tau_s})}}} & \end{cases}$$

SB_k *being the specific background for probeset k*

$$SB = T_b(\log_2^{PM_{k,i}} - \log_2^{MM_{k,i}}, i = 1,\ldots,n_k)$$

with contrast and scale tuning parameters $\tau_c = 0.03$ *and* $\tau_s = 10$

Equation 3 - IM background method computation

The adjusted PM intensity is obtained by subtracting the corresponding IM from the observed PM intensity.

Normalization

The purpose of this step is to adjust data for technical variation, as opposed to biological differences between the samples. There will always be slight discrepancies between the hybridization processes for each array and these variations tend to lead to scaling differences between the overall fluorescence intensity levels of various arrays. For example the quantity of RNA in a sample, the amount of time for which a sample spends hybridizing or the volume of a sample can all introduce significant variance. Even subtle physical differences between arrays or between the scanners used to read arrays can have an effect. Put simply, normalization ensures that when

comparing expression levels of different arrays that we are, as much as is possible, comparing like with like (Watson, et al., 2007).

With reference chip

Scaling: For this normalization method, a baseline array is chosen and all the other arrays are scaled to have the same mean intensity as this array. This is equivalent to selecting a baseline array and then fitting a linear regression, without an intercept term, between each array and the chosen array. Then, the fitted regression line is used as the normalizing relationship. One modification is to remove the highest and lowest intensities when computing the mean, that is, a trimmed mean is used. Affymetrix removes the highest and lowest 2% of the data. Affymetrix has proposed using scaling normalization (Robinson and Oshlack) after the computation of expression values, but it may also be used on probe-level data. Another modification is to use a target mean value in place of the mean intensity on the baseline array.

Non-linear methods: Methods that perform non-linear adjustments between arrays have been proposed and tend to out-perform linear adjustments such as the scaling method. Numerous non-linear relationships have been proposed including cross-validated splines (Schadt, et al., 2001), running median lines (Li and Hung Wong, 2001), and loess smoothers (Bolstad, et al., 2003). For a typical implementation, the normalizing relationship is fitted using a rank-invariant set of points, that is, a set of points that has same rank ordering on each array.

Without reference chip

Quantile normalization: The goal of quantile normalization is to impose the same empirical distribution of intensities to each array. A quantile-quantile plot will have a straight diagonal line, with slope 1 and intercept 0, if two data vectors have the same distribution. This kind of quantile normalization method is not the only normalization method based upon quantiles (Workman, et al., 2002) (Amaratunga, et al., 2001).

Cyclic loess: The cyclic loess method is a generalization of the global loess method (Yang, et al., 2002), where Cy5 and Cy3 channel intensities are normalized on cDNA microarrays by using MA-plots. When dealing with single-channel array data, it is pairs of arrays that are normalized to each other. The cyclic loess method normalizes intensities for a set of arrays by working in a pairwise manner. With only two arrays, the algorithm is identical to that in Yang et al. (2002b). With more than two arrays, only part of the adjustment is made. In this case, the procedure cycles through all pairwise combinations of arrays, repeating the entire process until convergence. One drawback is that this procedure requires $\boldsymbol{O(n^2)}$ loess normalizations although usually only one or two complete cycles through the data are required.

Contrast normalization: The contrast normalization method (Bolstad, et al., 2003) is another generalization of the methods described by Yang et al. (2002b). In brief, the data are transformed to a set of contrasts, a non-linear MA-plot normalization is performed, and then a reverse transformation is applied. It requires only ***O (n)*** loess normalizations, which is considerably fewer than with the cyclic loess method. As with the cyclic loess method, a subset of the data can be used to fit the loess curves, leading to considerably reduced running times. One way that the subset may be chosen is to use a rank-invariant set of probes.

Summarization

It is the process of combining the multiple probe intensities for each probeset to produce an expression value. There are a number of different ways that this can be achieved, but the end result is always a single expression value for each gene on each chip.

Analysis packages

Most of the times, the preprocessing methods described above are combined in single analysis packages such as RMA, GCRMA (GC Robust Multichip Averaging) and others. They provide quite general facilities for computing expression summary values. In particular they allow most background adjustment, normalization, and summarization methods to be combined. The choice of preprocessing method can have enormous influence on the quality of the ultimate results (van de Wiel, et al., 2010). It is important to assess different methods before proceeding to more downstream analysis.

Probeset-level analysis

Although some researchers have focused their statistical analyses on probes, many of them still prefer to work at the probeset-level, where data can be analyzed in different layers (individual analysis, geneset[8] analysis, co-expression analysis and clustering) (Berger, et al.).

[8] Geneset refers to a group of genes.

Detecting differentially expressed genes

Single Gene Approach

After removing the bad quality data we are left with reliable data. As we have already seen, good quality data can be further filtered so that only the genes that show some changes in the expression during the experiment are preserved in the dataset (Cooper and Shedden, 2003). Often the differentially expressed or otherwise interesting genes are stored as simple lists of gene names, i.e., genelists[9].

There exist many statistical test methods based on number of sample groups and replicates (Table 2). In this thesis, we will briefly and only cover some of the popular existing algorithms and present novel approaches we developed during the course of this thesis work. Among these, are statistical variants of t-tests and fold change.

Sometimes it is sufficient to get information about genes that are either under- or over-expressed during the experiment (Santarius, et al.). For example, we might be interested in genes that have an elevated expression because of a drug treatment. Such genes are most easily found by simple filtering. Simple filtering can even be used for experiments, where there are no replicates (Lu, et al.).

Also, in single gene differential expression analysis, if the log-transformed data is used, the over-expressed genes have an expression above zero and under-expressed genes have an expression below zero. However, often the experimental errors are of the order of two, and for finding the over- and under-expressed genes the cutoffs of -1 and 1 are used. All the genes which fulfill the filter criteria are saved in a new gene list. The filtering by absolute expression change is not always the optimal method for finding the differentially expressed genes, because the information about the reliability of the expression change is lacking (Lai). In the absence of replicates, there are still a few statistical methods that can be used, if we want to have statistical significance values for expression changes. However, using such methods carries the risk that the most unstable probes or mRNAs are identified as differentially expressed (Stylianou, et al., 2008).

However, there exist clear advantages of statistical **t-test** over fold change threshold for selecting differentially expressed genes (McCarthy and Smyth, 2009) (Murie, et al., 2009).

These advantages include among others:

[9] Genelist refers to an assumed list of genes belonging to the same group for a specific purpose.

- Incorporation of variation between measurements,
- Estimation for error rate,
- Detection of minor changes,
- Ranking of differentially expressed genes.

Table 2 - Summary of algorithms which may help users to select statistical testing methods based on the number of sample groups and replicates and potential correction methods.

Sample Groups	Sample Replicates	Basic Statistical Method	Multiple Testing Procedure	Gene (feature) Limit Options	Main result files
1	Any	None	None	None	None
2	< 3 in either group	Average values of each gene (feature) within group Calculate fold change of each gene between groups	None	Fold Change Total number of genes User gene list	Gene list Fold changes Expression values Heatmap of top genes
	≥3 in both groups	two-sample Welch t-test (unequal variances) two-sample t-test (equal variances) standardized rank sum Wilcoxon test paired t-test Options for Raw/Nominal p-value calculation: *Parametric* *Permutation* Options for Side/Rejection Region: abs, upper, lower	Bonferroni single-step FWER Holm step-down FWER Hochberg step-up FWER Sidak single-step FWER Sidak step-down FWER Benjamini & Yekutieli step-up FDR Benjamini & Hochberg step-up FDR Storey q-value single-step pFDR Westfall & Young maxT	Fold Change Limit to *Total number of genes* *adjusted p-values* *raw p-values* *test statistics* User gene list	Gene list Fold changes Statistic Raw p-values Adjusted p-values Expression values

			permutation FWER Westfall & Young minP permutation FWER		
3	≤3 in any group	Calculate percentile of standard deviation (SD) of each gene cross all samples Select genes by a SD percentile cutoff		Standard Deviation (SD) Total number of genes User gene list	Gene names Standard Deviation Expression values
	≥3 in all groups	F-test Block F-test Options for Raw/Nominal p-value calculation: Parametric Permutation Options for Side/Rejection Region: abs, upper, lower	Bonferroni single-step FWER Holm step-down FWER Hochberg step-up FWER Sidak single-step FWER Sidak step-down FWER Benjamini & Yekutieli step-up FDR Benjamini & Hochberg step-up FDR – selected Storey q-value single-step pFDR Westfall & Young maxT permutation FWER Westfall & Young minP permutation FWER	Total number of genes User gene list	Gene names Statistic value Expression values

Single Gene Differential Expression Analysis

To assess the confidence of an experiment analysis, we need to determine the statistical significance of the Up/Down-regulation of the gene. Significance can only be assessed, if replicate measures of the same gene were performed during the experiment (Siddiqui, et al., 2006). If the gene expression has been measured for control and treatment, but only treatment has been replicated, the experimental error can be crudely estimated from the variation between the replicates (Schneider and Roossinck, 2001). The standard deviation is determined for every gene, and the expression change is compared with the standard deviation. The more the change exceeds the standard deviation between replicates, the more significant the gene is (Bozinovic, et al.). However, this method does not allow for the calculation of p-values.

Parametric tests

When conducting this kind of classical **t-tests**, researchers assume that data fits a normal distribution (Goddard and Hinberg, 1990) (however this assumption vanishes when dealing with a larger number of conditions [n>30] and Kolmogorov-Smirnov (Young, 1977) or Shapiro-Wilk (Henderson, 2006) tests remain formal tests for normality Q-Q plots), tests are independent and gene expression levels have equal variance across conditions. Additionally, the following conditions should clearly be considered when applying classical t-tests (Hoaglin, et al., 2000):

- **One condition with multiple replicates (one-sample t-test):** Up- and down-regulated genes can be more effectively found, if the chips are replicated. In such cases, the one-sample t-test for the deviation from zero (expected mean) can be performed. This indicates that new chip technology coupled with an appropriate number of replicates can produce reliable data even for very small expression changes.

$$t = \frac{\sum X_i}{\sqrt{\frac{n.\sum X_i^2 - (\sum X_i)^2}{n-1}}}$$
$$X_i = V_2^i - V_1^i$$

Equation 4 - Classic t-test computation

- **Two conditions with multiple replicates (two-sample t-test):** The two-sample t-test looks at the mean and variance of the two distributions (say, control and treatment chip log ratios), and calculates the probability that they were sampled from the same distribution. The t-test can be applied successfully in

situations, where both the control and treatment have been repeated. After calculating the t-test p-values for the replicated genes, the ones with the lowest p-value are the ones which most significantly differ between two conditions, say control and treatment.

- **Multiple conditions with multiple replicates (ANOVA):** In this case, the fundamental idea behind analysis of variance (ANOVA) is that, given an appropriate experimental design, variability in the quantity being measured (gene expression) can be partitioned into various identifiable sources. The assumed sources of variability will include the experimental factors, as well as random noise. There are a number of assumptions which must be made before ANOVA can be applied; deviation from the assumptions will lead to misleading or inaccurate results. These assumptions include the independence, normality, and uniformity of variance of the errors. Importantly, we assume (or at least hope) that the linear model we select is adequate to describe the data. In some situations, there are adjustments that can be made to reduce violations of these assumptions, such as log transformation of the data, and there are well-established methods for diagnosing the quality of the results. However, for microarray data, there are likely to be genes for which the assumptions are valid and others for which they are not. The situation may improve for some genes and worsen for others after correction attempts. The use of non-parametric methods can greatly reduce the reliance on prior assumptions about the data, but also incur a loss of power.

Non-parametric tests

Instead of parametric t-test, which assumes that the expression values are normally distributed non-parametric tests like Mann-Whitney U test (two groups) or Kruskal-Wallis test (two or more groups) can be applied, especially if the expression values are not normally distributed.

Gene expression values don't follow a normal curve, but the p-values from a standard t-test aren't far away from truth when the distribution is moderately asymmetric (Hollander, et al., 1999). However the t-test does fall down badly when there are outliers (values more than twice as far from the mean as all the others). For this reason, when doing a t-test, it is wise to confirm that neither group contains outliers. In practice, it often happens that genes detected as different between groups, are actually expressed very highly in only one individual of the higher group. Probably the best way to avoid the problem with outliers is to use robust estimates of mean (trimmed mean) and variability (mean absolute deviation). However no theoretical

distribution is known for these; and their significance levels (p-values) would have to be computed by permutations.

Multiple testing problem

P-Values and False Discovery Rates

If the aim of the microarray study is to select a few genes for more precise study, then the goal is an ordered list of genes, most of which are really different (true positives). Another way to say this is that the expected number of false positives is some reasonable fraction of the genes selected. This goal leads naturally to specifying the false discovery rate (FDR) (Yarden and Sliwkowski, 2001) for a list, rather than significance level (false positive rate). The FDR is the expected fraction of false positives in a list of genes selected applying a particular statistical procedure. If applying this procedure in many experiments would give gene lists including twenty per cent false positives, on average, then the procedure's FDR is 20%. The FDR is distinct from the false positive rate (FPR), which is the rate at which truly unchanged genes appear as false positives.

Multiple Testing P-Values and False Positives

Suppose you compare two groups of samples drawn from the same larger group, using a chip with 20,000 genes on it. On average 1000 genes will appear 'significantly different' at a 5% threshold. For these genes, the variation between samples will be large relative to the variation within groups due to random, but uneven allocation of the expression values to the treatment and control groups. Therefore the p-value appropriate to a single test situation is inappropriate to presenting evidence for a set of changed genes (Westfall, et al., 1993). Statisticians have devised several procedures for adjusting p-values to correct for the multiple comparisons problem.

Multiple testing correction methods

There are two ways of controlling multiple testing error rates (Gordon, et al., 2007):

- Family-wise error rate (FWER): used to correct for the probability of at least one **type I** error (Methods: Bonferroni-Holm, Benjamin-Hochberg …)
- False Discovery Rate (FDR): used to correct proportion of **type I** errors among rejected hypotheses (Methods: fdr…)

One way to improve such statistical analyses is to integrate biological information in the design of these analyses. During the course of this thesis, we have used the relationship between the level of gene expression and variability. Using this biological information, we have proposed to integrate the information from multiple genes to get a better estimate of individual gene variance, when a small number of replicates are available, to increase the power of the statistical analysis. We described a strategy named the "Window t test" (Berger, et al., 2008) that uses multiple genes which share a similar expression level to compute the variance which is then incorporated a classic t test. The performances of this new method were evaluated by comparison with classic and widely-used methods for differential expression analysis (the classic Student t test, the Regularized t test (reg t test), SAM, Limma, LPE and Shrinkage t). In each case tested, the results obtained were at least equivalent to the best performing method and, in most cases, outperformed it. Moreover, the Window t test relies on a very simple procedure requiring small computing power compared with other methods designed for microarray differential expression analysis.

However, the sensibility of microarray data analysis tools and methods is strongly limited by the weakness of the estimation of variance because the number of replicates is generally low and variance heterogeneity high. The above mentioned methods and their variants have tried to increase this sensitivity by improving the estimation of variance. However, these methods are generally evaluated and/or tested on artificial "spike-in" or simulated data.

Consequently, the ability of these methods to better estimate variance is tested only on technical or modeled variances, and not on biological variances.

During the course of this thesis, we co-developed and co-authored a novel approach to circumvent this limitation by evaluating existing single gene statistical methods on actual biological data. This limitation is mainly because the use of actual data does not allow for the definition of the unambiguous "truth" to identify true and false positives.

In this work (De Hertogh, et al., 2009), we showed that the Shrinkage *t* test (close to Limma) was the best of the methods tested, except when two replicates were examined, where the Regularized t test and the Window t test performed slightly better.

The benchmark method proposed here differs from other approaches published, as actual biological and experimental variability is preserved. The obtained Mean – Standard deviation relationships confirm that the variance structure of the data we studied is closer to biological data than that of spike-in or simulation studies. One other advantage of the method lies in the fact that virtually all parameters can be fine-tuned, allowing researchers to assess those methods which are truly suited for their particular approaches.

We applied the benchmark to a set of published methods. The results show better performances for the Shrinkage *t* test, except when there are only two replicates, where the Regularized *t* test and Window *t* test perform better (Figure 6).

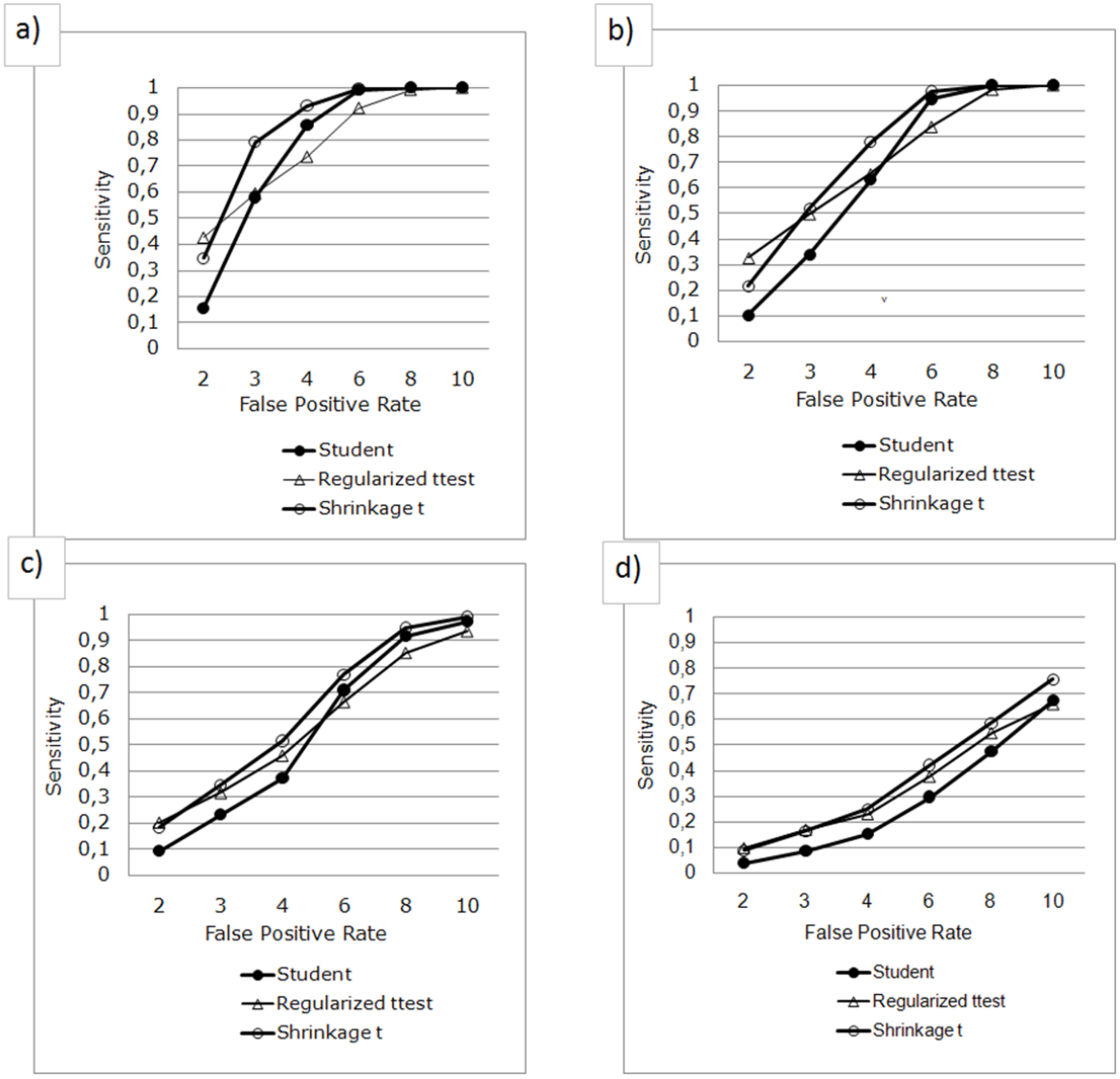


Figure 6 - Shrinkage t performance.

The figure above show the absolute performances of different tested methods. Absolute sensitivity versus number of replicates for classic Student's t test, Shrinkage t and Regularized t test. From a to d: Runs 1 to 4 (DEGs increasingly difficult to find).In this study, the Shrinkage t performs best overall, except for 2 replicates.

Single Gene vs Geneset Differential Analysis Dilemma

The double edged problem of multiple testing and the week variance estimation (due to the limited number of replicates) hamper microarray data analyses.

Departing from normality and variance heterogeneity between genes and between experimental conditions for a given gene can decrease the confidence of statistical tests. Moreover, data has shown that a non-trivial mean-variance relationship benefits to methods analyzing groups of genes. This relies on the fact that n genes sharing similar expression levels also share more similar variances than n genes sampled randomly.

Geneset Differential Analysis

Several geneset analysis methods have been proposed to analyze the expression values of several genes related by known biological criteria (metabolic pathway, pathology signature, co-regulation by a common factor, etc.) at the same time and the cost of these methods allows for the use of more values to help discover the underlying mechanisms.

However, geneset differential expression analysis is a far more complex task than the single gene analysis of expression changes. The diversity of the biological criteria involved and prior definition of the genesets has an impact on the mathematical properties of the expression subsets to be analyzed. Furthermore, the diversity of available analysis procedures, each based on specific strategies and null hypotheses have to address these properties. Thus, design of the analysis strategy is not that simple. Current methods involve over-representation analysis (ORA), and functional class scoring (FCS). The FCS methods rely either on 2-step (post-hoc) or global strategies (using raw data).

Few methods have tried to solve this issue. Only the FAERI methodology (Berger, et al.) tried to solve it. FAERI being a global methodology tailored from a 2-factor ANOVA procedure by a 2-step reduction of the data, and is evaluated with respect to the self-contained null hypothesis (using label sampling or random data).

FAERI thus constitutes an improvement over all tested methods, and turns out to be the optimal method for testing question: “Which known genesets are associated with different expression profiles under the two conditions compared?”

A real-world example of analysis is reported for three datasets on cellular response to hypoxia. The results obtained from analysis of the E-MEXP-445 dataset illustrate that FAERI (evaluated using permutations) is able to detect relevant results when a strict cut-off is used, compared to the other methodologies. The genesets detected by

FAERI are related to three categories, respectively metabolic perturbations, hypoxia signaling/response, and hypoxia-related pathologies. The results of other global methods confirm these results, and posthoc methods fail to detect any significant genesets. The results provided by FAERI from analysis of the two additional datasets reveal its ability to detect the same sets using several related datasets. Focusing on the top list obtained with several datasets, we showed that

FAERI not only detects more sets, but also presents a larger intersection of the results.

We proposed to score this last assessment on the whole list of sets by computing the Pearson (Pearson, 2008) correlation coefficient on the ranked list of genesets, for each method, using pairwise comparisons between the three datasets.

Among all methods tested, FAERI provides the best correlated results between related datasets; regardless of the source of geneset definition used (each category of the MSIGDB[10] was used for this purpose). FAERI thus outperforms all methods tested for its ability to attribute similar scores to each geneset from several datasets.

Microarray noise issues

One of the major difficulties in deciphering high throughput gene expression experiments comes from the noisy nature of the data.

Microarray data are commonly perceived as being extremely noisy because of many imperfections inherent in the microarray technology (Barrett, et al., 2009).

To correctly interpret the gene expression microarray data, it is crucial to understand the sources of the experimental noise. There are two principal sources of noise in microarray experiments: biological noise and technical noise. Biological noise consists of variation among patients and tumor locations, variation in the cellular composition of tumors, heterogeneity of the genetic material within tumor due to genomic instability.

Technical noise consists of differences in sample preparation and experiment variables which include non-specific cross hybridization, differences in the efficiency of labeling reactions and production differences between microarrays.

A recent study suggests that, contrary to popular belief, the random fluctuations of gene expression signals caused by technical noise are quite low and the effect of such

[10] Molecular Signatures Database, *www.broadinstitute.org/gsea/**msigdb**/index.jsp*

fluctuations on the results of statistical inference from Affymetrix GeneChip microarray data is negligibly small (Klebanov and Yakovlev, 2007).

If the noise is consistent and reproducible it can be filtered from the data and some false positives can be eliminated. The noise derived from microarray experimental techniques is reproducible and its boundaries can be modeled by statistical algorithms. However, biological noise cannot be completely corrected but it can be accounted for and reduced with statistical methods using replicates of the conditions or other approaches.

This approach is often costly for many researchers and sometimes impossible for others given limited resources; thus, appropriate methods which increase accuracy at no additional cost, are needed. One inexpensive source of microarray replicates comes from prior studies: to date, data from hundreds of thousands of microarray experiments are stored in the public repositories. Although these data assay a wide range of conditions, they cannot be used directly to inform any particular experiment and are thus ignored by most differentially expressed gene methods.

Another promising approach is the one we proposed in this research study.

This approach consists of a tool that integrates biological information and combined microarray gene expression data, to reduce the background noise which limits the interpretation of microarrays experiments.

Thesis structure

This thesis consists of a further six chapters structured and detailed as follows:

Chapter II presents the genesis of this work (PathEx web tool background), context and motivation in the field of microarray analysis. We reviewed the position of the current research with respect to known near competitors.

Chapter III detailed PathEx rationale, design steps, implementation and testing.

Chapter IV covers PathEx results and critical assessment of PathEx by discussing its efficiency/effectiveness when applied to different partners published studies. Three validations (two on meta-analysis of metastasis and another on critical mediators of atopic dermatitis pathways) are discussed.

Chapter V highlights further directions of current research, presenting a planned project of creating a broad microarray pipeline analysis tool comprising PathEx, Pegase[11], FAERI and gVIz analysis tools. It also discusses in details the development of gViz, a tool implemented with two-fold interest notably prediction and validation of gene groups via the use of annotation information from PathEx database component.

Chapter VI discusses the main conclusions of this research by summarizing the work completed, by discussing how the research questions have been answered. Also, the contributions to knowledge achieved through the application of PathEx are discussed. The chapter rounds up with some final remarks.

[11] Pegase is an old version of FAERI method; in Pegase ANOVA-2 is not implemented.

Chapter II

Thesis genesis

Aim of the study

One of the objectives of microarray experiments is to identify genes that are differentially expressed in biological samples under different conditions (e.g., disease vs. control). The samples may come from tissues extracted from different organs or parts of the same organ (e.g., different brain regions). In this case, we may be able to discover differentially expressed genes in each organ/organ part and how disease may affect each organ/organ part at the gene expression level.

Although there has been a trend whereby many researchers widely use microarray technologies, less is done computationally to interpret and validate biological hypotheses formulated from inherent investigation results. Continued microarray data deposit and revision of genome annotations are important to supplement previously submitted microarray metadata. While the advent of microarray technologies and an increasing number of analysis methods present an opportunity to better understand life mechanisms, exploitation of microarray data and the choice of analysis methods remain challenges.

Also, considering that technologies like microarrays remain prohibitively expensive for researchers with limited means to order their own experimental chips, it would be beneficial to re-use previously published microarray data.

For certain researchers interested in finding gene groups (requiring many replicates), there is a great need for tools to help them to select appropriately datasets for analysis. These tools may be effective, if and only if, they are able to re-use previously deposited experiments or to create new experiments not initially envisioned by the depositors. However, the generation of new experiments requires that all published microarray data be completely annotated, which is not currently the case.

We present, in this thesis, a novel approach (PathEx) which, taking into consideration the above facts, has contributed to solve various issues related to microarray analysis.

PathEx is, a human-focused web solution built around a two-component system: one database component, enriched with relevant biological information (expression array, omics data, literature) from different sources, and another component comprising sophisticated web interfaces that allow users to perform complex dataset building queries on the contents integrated into the PathEx database.

The idea behind the development of PathEx is three-fold:

First, while conducting a benchmark of different microarray statistical analysis methods (De Hertogh, et al., 2010), it was found that some methods focusing on finding gene groups might require many replicates. For a researcher considering conducting a microarray analysis, one consideration should be taken into account: the dataset of interest. As single microarray dataset usually contains a large number of genes (hundreds or thousands) but the number of observations is much lower – generally tens or up to a few hundred at the very most; which makes it very difficult to extract reliable biological information from a single dataset. However, it only makes sense to combine data sets if the questions are the same, or, if some aspects of the experiments are sufficiently similar that one can hope to make better inference from the whole than from the experiments separately. Several studies have proved that combining data from multiple experimental studies can improve biological prediction results, even in the case where the scientific focus and experimental conditions of the individual microarray studies differ from one another (Zhu, et al., 2008) (Ivliev, et al., 2008) (Menssen, et al., 2009).

Second, to use a combinational approach of enriched (with different types of biological information) microarray data from public microarray repositories to generate focused datasets. These specific datasets, which may be used in co-expression analysis by organizing genes into different functional groups based on the principle that genes belonging to the same functional groups or pathways will have similar expression profiles over a range of experimental conditions. This is true because biologists often already knew a subset of genes involved in a specific genes group of interest (pathway, disease state…) and may wish to discover other genes that can be assigned to the same group. As such, the co-expression analysis approaches are more suitable. As such, supervised dataset selection algorithms will tend to assign group memberships to genes that correspond well to the true underlying biological phenomena. One major drawback of these approaches is that annotation is learned directly from the expression data without taking advantage of the often available predefined annotation information contained in different biological databases/databanks. As a result, co-expression analysis approaches can generate groups of genes that do not correspond well to the true underlying biological pathways.

Third, features provided in this approach, allow it to generate any kind of datasets, with an added value of enriching their meta-data with biological annotations from different sources of biological data, reducing at the same time a portion of microarray data noise. The outcome tool integrates easily any statistical analysis pipeline tools.

State of the art

We propose here a novel web tool that combines information from microarray data, the literature and "omics" technologies. Its main objective is to allow for instantaneous selection and generation of datasets of interest by drawing relevant samples files from major publicly available microarray repositories and using simple but biologically meaningful keywords to query the underlying database. PathEx provides biologists (with no or limited pre-knowledge of the structure and organization of the microarray data) with an intuitive web interface to generate datasets for validation of existing studies, discovery of new phenomena or complementation of hypotheses regarding phenomena only partially understood.

Many researchers must often manually retrieve or use certain tools available to retrieve microarray data from public repositories. However, such tools are most often limited to pre-knowledge of the structures and formats of the deposited microarray data.

Several tools proposed are mainly either retrieval tools (Microarray Retriever (MaRe) (Ivliev, et al., 2008)) or full integrated but manufacturer-oriented analysis tools (combining retrieval and analysis tools: EzArray (Zhu, et al., 2008) and SiPaGene (Menssen, et al., 2009)).

However, none have the enhanced ability to allow researchers to automatically select data of interest by focusing on certain biological factors that were not necessarily those provided in the microarray metadata.

Unlike existing tools, the power of PathEx is its fast processing capability made possible through local storage of all of the data (to avoid the sequential downloading policies and bandwidth limitation associated with most microarray repositories). PathEx also remains unique in that it acts as a point of integration of fully re-organized information from public sources. Furthermore, PathEx is not bound to any microarray manufacturer or type. This allows for the datasets selected by PathEx to be analyzed by any platform associated analysis method.

Chapter III

Design

PathEx rationale

As PathEx does not aim to be yet another microarray retrieval tool and the main goal was to develop a novel concept to offer less exploited opportunities for the analysis of deposited microarray data. Deposited microarray data comes with description files (though these files are sometimes incomplete). These metadata files do however contain some key information that can be used to link the microarray data to other biologically related information. We propose here a system that uses this identification metadata to link microarray data to other biological concepts such as *Genes*, *Proteins*, *Metabolic Pathways* and *the Literature*. By further characterizing previously deposited microarray data; we provide researchers with new opportunities to select interesting datasets by simply using meaningful biological criteria to query the underlying PathEx database.

PathEx data sources

In recent years, the number of biological data sources, and the quality and the quantity of information available to life sciences researchers have increased at a very fast rate. Such data sources are now essential to some of them with limited financial means. However, for computational tasks involving an answer to a biological question, researchers often spend much of time and effort to (a) manually use the interfaces of biological data sources, (b) extract and import data into their own environments, (c) relate various disconnected information found in the extracted data, (d) refine the search criteria, and (e) repeat the whole process from the beginning.

Our goal is to provide with researchers an integrated web software system which accesses multiple biological data sources, integrates the retrieved data, and create knowledge to reduce noise emanating from raw microarray data.

Mainly PathEx data comes from two categories of sources:

1. Microarray data which come mostly from two major microarray repositories National Center for Biotechnology Gene Expression Omnibus (NCBI GEO) (Barrett and Edgar, 2006) and European Bioinformatics Institute Array Express (EBI AE) (Rocca-Serra, et al., 2003) (Brazma, et al., 2003) (Parkinson, et al., 2005) (Parkinson, et al., 2007).
2. Other biological information used to further characterize that microarray data is mainly taken from major omics databases/databanks. These data mainly

concern gene annotations from EntrezGene (Maglott, et al., 2005), Kyoto Encyclopedia of Gene and Genomes (KEGG) (Ogata, et al., 1999) (Kanehisa and Goto, 2000), ENSEMBL (Hubbard, et al., 2002) (Birney, et al., 2004) (Hubbard, et al., 2005) (Birney, et al., 2006) (Hubbard, et al., 2007) (Flicek, et al., 2008) (Hubbard, et al., 2009), H-InvDB (Yamasaki, et al., 2009), Vertebrate Genome Annotation (Vega) (Ashurst, et al., 2005) (Wilming, et al., 2008), protein information from UniProt/Swiss-Prot (Boeckmann, et al., 2003), ENSEMBL, metabolic pathway information from KEGG Pathways, disease information from OMIM (Rashbass, 1995) and literature from PubMed (Guillaume, 1998).

The question is "How to effectively manage this huge and heterogeneous amount of data?"

In this thesis, we step by step describe how PathEx has provided researchers with an effective web application of advanced data management technology to biological data.

Biological data challenges

Every day, a large amount of biological data is produced and need to be organized, queried and reduced to useful scientific knowledge. Although data management technology can provide solutions to problems, in practice the data needs of biomedical research are not well served and existing data management technology is often challenged by the lack of stability, evolving nature, diversity and implicit scientific context that characterize biological data.

Biological data are broad and diverse. Biology encompasses many domains of knowledge. Each of domains those is concerned with overlapping or complementary entity types, and has its own terminology and data needs. Furthermore, the variety of experimental procedures yield related but not identical data. New bio-analytical procedures and progress in the science add another dimension of instability to biological data types.

Most of the information that the biological research is interested in is available in public reference databases and specialized private data sources. It is estimated that much of the biological data are in text form, and the rest resides in databases that range from indexed files, to relational and specialized formats.

Biological databases may contain primary data i.e., have their own data entry or submission policy, or secondary data i.e., built by integrating data from primary sources in which case their integrity depends on the constituent sources. As many of these data sources are non-standard and not well documented, accessing, integrating and sharing biological data becomes a challenge and an art.

Biological data management

A scientist may query a biological database, manually peruse the results of this query, find the data of interest, and use this data in a query of a second biological database. Performing manual multi-database queries of this sort can be a time-consuming process, especially when large amounts of data are involved, as is often the case in biological research. Therefore, a need exists for automated data integration from multiple databases.

Data materialized integrations (materialized integration) and database federations (virtual integration) are two approaches that have been developed to deal with this data integration. In materialized integration, data are combined from multiple data sources into one large database that houses the data of the individual databases. Materialized integration, which includes data format conversions, is performed at the time of the data materialized integration creation. Queries are performed on this integrated data source. In virtual integration, software modules are created to interact with the databases included in the federation. Data integration is performed at run-time at the time of the query, since the data must be obtained on the fly from all the separate databases involved in the query. Both approaches have several advantages and drawbacks that will be discussed later.

Integration issues related to biological data sources

Managing and integrating biological data collected by scientists and researchers from around the world can be very challenging for various reasons (the quantity of biological data, the large number of biological databases, the rapid rate in the growth of biological data, the overabundance of data types and formats, the variety of data access techniques, database heterogeneity, errors in biological data, and even the interdisciplinary nature of the field of bioinformatics).

Virtual integration or federation approach

The federation approach (Figure 7) to integrating data from heterogeneous databases does not require the creation of a new giant database to hold transformed versions of the data from constituent databases as in the data materialized integration approach. Interoperability does not require global integration. It is possible to develop software that can operate on multiple databases without solving the much harder problem of [globally] integrating those databases. In the federation approach, individual data sources act in a completely autonomous fashion as if they are not part of the federation. The federation is dependent upon its constituent data sources, but the individual data sources are in no way dependent upon the

federation, since they may not even be aware of the existence of the federation. To emulate a single data source in the federation approach, one or two types of software component are often used: mediators and translators.

A translator is responsible for tasks such as performing the data format conversions that are necessary for a particular source's data to be used in a federation query. For instance, a source may contain data stored as objects, but the translator may convert this object data into a relational table to be used by the federation. A translator is useful for resolving certain types of heterogeneity such as value heterogeneity.

A mediator makes a determination as to which data sources in the federation need to be involved via translators in a particular query. After translators return their results to the mediator, the mediator must perform integration of the data from the different sources.

Mediator and translator updating can be accomplished in different ways. For example, certain knowledge or federation standards may be hard-coded into the software components. It is also possible to have a separate federation database dedicated to holding syntactic and semantic knowledge of the various databases involved in the federation, and changes to data in this data-dictionary-like database could be used to periodically update translators and mediators. Trigger-like mechanisms could be used so that changes to this database could automatically update the affected translators and mediators. The ability to save historic data in this database could be essential so that if problematic data changes cause some of the mediators and translators in the federation to malfunction, it could be possible to restore earlier working configurations.

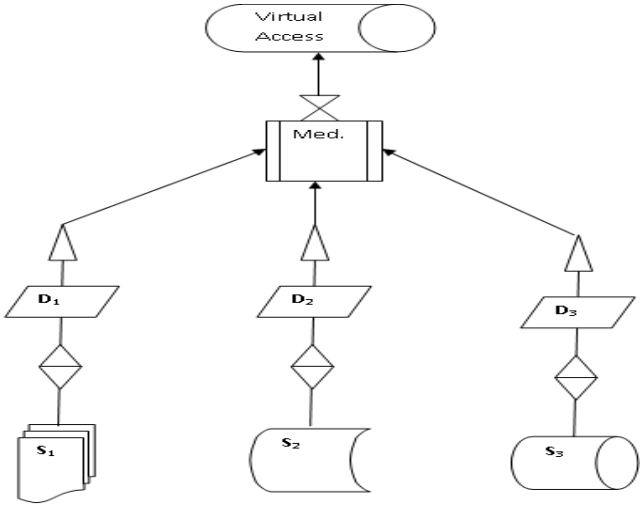


Figure 7 - Federation approach architecture

In federation approach, each source is queried separately and snapshots of interesting information are mediated in a virtual view.

There are several advantages to the federation approach. Federations do not require a global schema as is the case in typical data materialized integrations. Federated databases only require partial integration. A federated database integrates a collection of local data source systems by supporting interoperability between pairs or collections of the local data sources rather than through a complete global schema. The upkeep of enormous data materialized integration with its involved, global transformation issues is avoided in the federation approach. This is an important issue as biological databases continue to grow in number and in content. A principal advantage of an integration scheme based on mediators is that it allows the individual data sources to operate autonomously, but to function collectively as a federation. However, data materialized integrations don't actually impede the functions of their constituent databases, so in actuality data materialized integrations also allow constituent databases to function autonomously. In fact, federation queries and data retrieval are performed against autonomous databases. This is truly an advantage of federations, since they can perform queries against the original databases. Federations see data updates at run-time, so there is no delay from the time data enters a database to the time it is available to users, as is the case in data materialized integrations.

However, federations have certain disadvantages relative to data materialized integrations. To begin with, since data format conversions need to be done at run-time, federation query performance is slower. Additionally, queries must be performed against multiple data sources. For example, in case sources are databases, this can be a time-consuming operation, since a query may require a large cross-database join, and this join would probably be performed at a mediator rather than in one of the constituent databases. Such a join would be much faster in the case of a single-site database like a data materialized integration. Issues such as this have been addressed in query optimization in distributed databases. For example, if one table in one database contained a million rows and another table in another database contained ten rows and a joint need to be performed, of course it would make more sense to perform the joint at the first site. In a federation, a mediator would probably perform the joint, which would be time-consuming. However, perhaps a mediator could work in conjunction with an existing DBMS to perform a joint by sending data from another database to that database so that the joint will be performed at one of the databases rather than in the mediator.

Materialized integration approach

In the materialized approach (Figure 8) to data integration, data from individual data sources is usually transformed into a common format and then stored in a large single database called a data materialized integration. In this case, data materialized integration is a single database that is constructed by physically consolidating a collection of data sources into an integrated whole… Thus, a materialized integration transforms a set of heterogeneous database s into a single, homogeneous database, and data stored in this way can be queried using the standard query tools of the DBMS (database management systems). The transformation is performed by a set of translators, each of which must convert a data source to a form that is compatible with the materialized integration. This conversion process can be complex, because source heterogeneity can manifest itself in a number of different aspects. The various types of heterogeneity (value and semantic) and other issues must be confronted during this integration phase that occurs when the data materialized integration is constructed.

Data materialized integrations are typically read-only, since they draw their data from existing biological databases and may do various data transformations and integration steps that make it impossible to write changes to the base databases via the data materialized integration. Writing data to the data materialized integration itself would not make sense since the data materialized integration is periodically updated from the base data sources and any data written to the data materialized integration would be lost when the data materialized integration was updated.

To construct a data materialized integration, monitoring systems must be implemented on the local databases to actively detect changes to them, and to propagate the changes to the data materialized integration. These changes are transformed into updates to the data materialized integration by a data integrator subsystem.

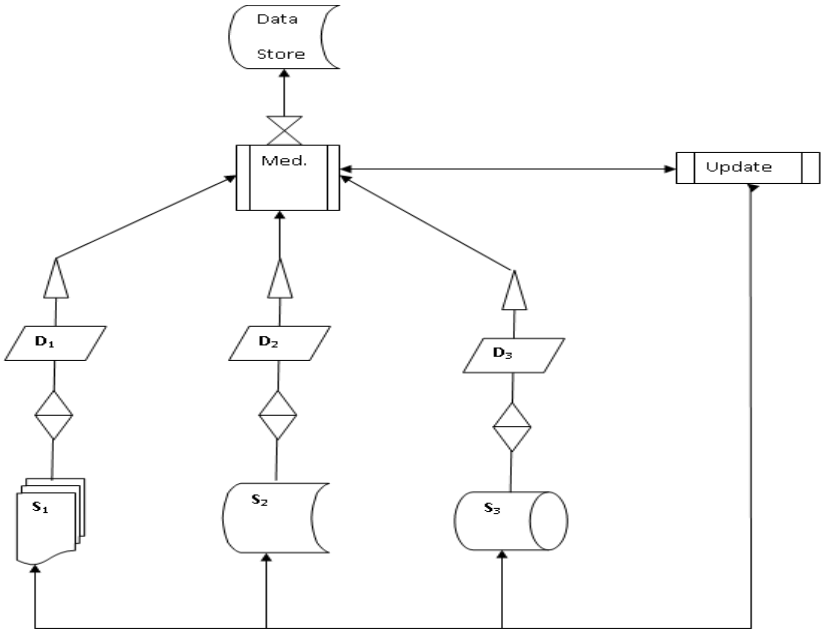


Figure 8 - Warehouse approach architecture

In this approach interesting information is imported, mediated and stored locally.

The advantages of materialized integration are that queries can be executed rapidly because all the data are located in one place, and the end user sees a homogeneous, integrated data source. Its disadvantages are that the integration of updates and revisions from the data sources into the materialized integration can be difficult and time consuming.

By contrast, the distributed approach makes updates available instantly.

In addition, as the number and size of biological data sources increase, the warehousing approach may not be able to keep pace because it becomes more and more time consuming to build large materialized integrations and to keep them up to date.

Moreover, the storage and computing resources required to maintain the materialized integration become increasingly expensive. The amount of knowledge required about local schemas, how to identify and resolve heterogeneity among the local schemas, and how changes to local schemas can be rejected by corresponding changes in the global schema are major problems with this approach because of the complexity of a global schema.

Integration efforts that depend on making local copies of data (possibly modified in a variety of ways) from other data sources must contend with a number of factors that will make their task increasingly difficult. Those factors including:

1. a proliferation of independently administered biological databases containing relevant data;
2. the absence of a clear domain boundary for data of interest to the potential users of such integrated data products (implying that there is no obvious limit to the set of course data sources that should be integrated);
3. and the accelerating rate of data production that will preclude manual intervention in the integration process (implying that all the information required to cross-reference data from two or more source databases must be present in or derivable from, those data sources).

PathEx integration approach

Contrary to materialized or virtual approaches, PathEx is built on a hybrid approach which makes the underlying PathEx database be read/written since data changes (updates) are made directly to this database. In this approach, we combined the advantages of both materialized and virtual integration approaches (Figure 9). PathEx combines the better of these two approaches and delivers a data integration platform that offers data federation as an integrated module. This made it easier, quicker, and cheaper to apply both data federation and data integration capabilities to different biological data challenge we encountered.

In this approach, where only virtual approach is possible, data drawn by the mediators are captured, converted and stored into a PathEx standard format to be later used in the relational database. This is mainly applied on proprietary data such those from KEGG and Array Express. While a data source can be physically copied, appropriate integrators were developed and sources physically imported before any conversion.

With this approach, we ended up with a single system, multipurpose environment PathEx, instead of multiple, independent tools, each with their own architecture, metadata, semantic model, and functionality.

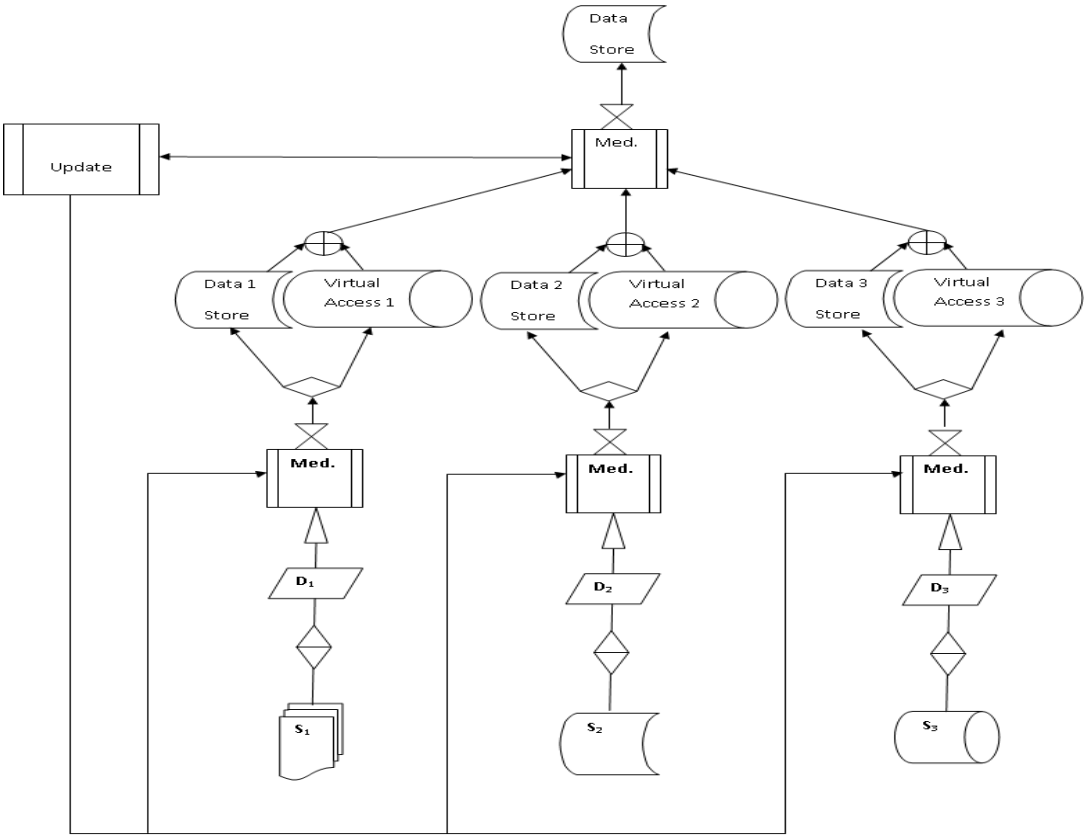


Figure 9 - PathEx data integration approach architecture

PathEx mixes the virtual and data warehouse approaches due to technical constraints and access limitation of some sources.

PathEx system architecture

The PathEx architecture is divided into three main components (Figure 10): The Processing Logic, The Contents Logic and The Navigator Logic. The Processing Logic has four interdependent utilities (Data Mining Utility, Integration Utility, Query Handler Utility and Updater Utility), The Contents Logic has two storage approaches (Database and Files Repository) and The Navigator Logic has several interfaces (Query Settings, Dataset Builder, Dataset Cart and Global Datasets Manager).

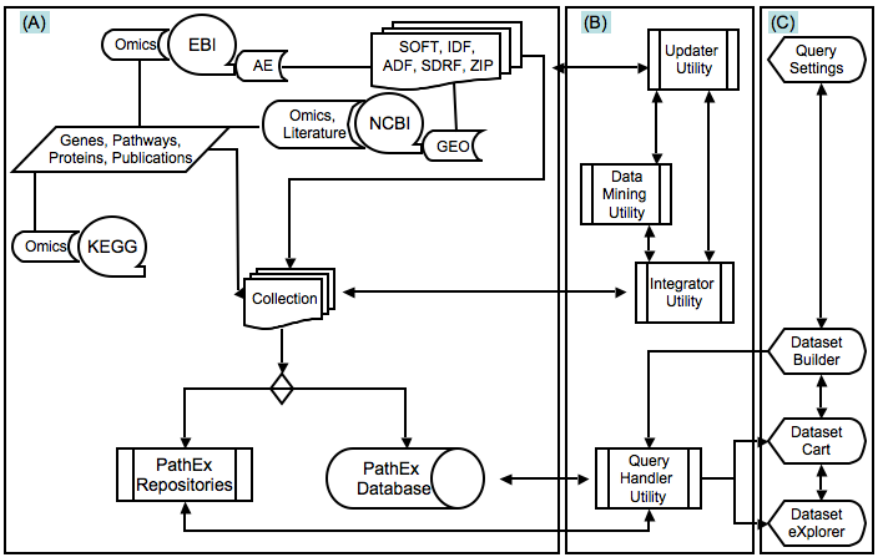


Figure 10 - PathEx system architecture

PathEx data processing logic

The PathEx Processing Logic is responsible for:

1. Federation of:
    a. Basic microarray data (platforms, experiments and samples) and associated sample raw data from GEO Simple Omnibus Format in Text (SOFT, http://www.ncbi.nlm.nih.gov/geo/info/soft2.html#SOFTformat) files (Figure 11) and Array Express MicroArray (Figure 12) and Gene Expression tab (MAGE-TAB) (Rayner, et al., 2006)) files,
    b. Additional reviewed microarray metadata, not primarily envisioned by the experiment owners (biological tags: sex, tissue, organ) and
    c. Biological information (genes, proteins, metabolic pathways and literature information),
2. Remote change tracking, monitoring and updating whenever required,
3. PathEx user and query management and
4. PathEx database integration.

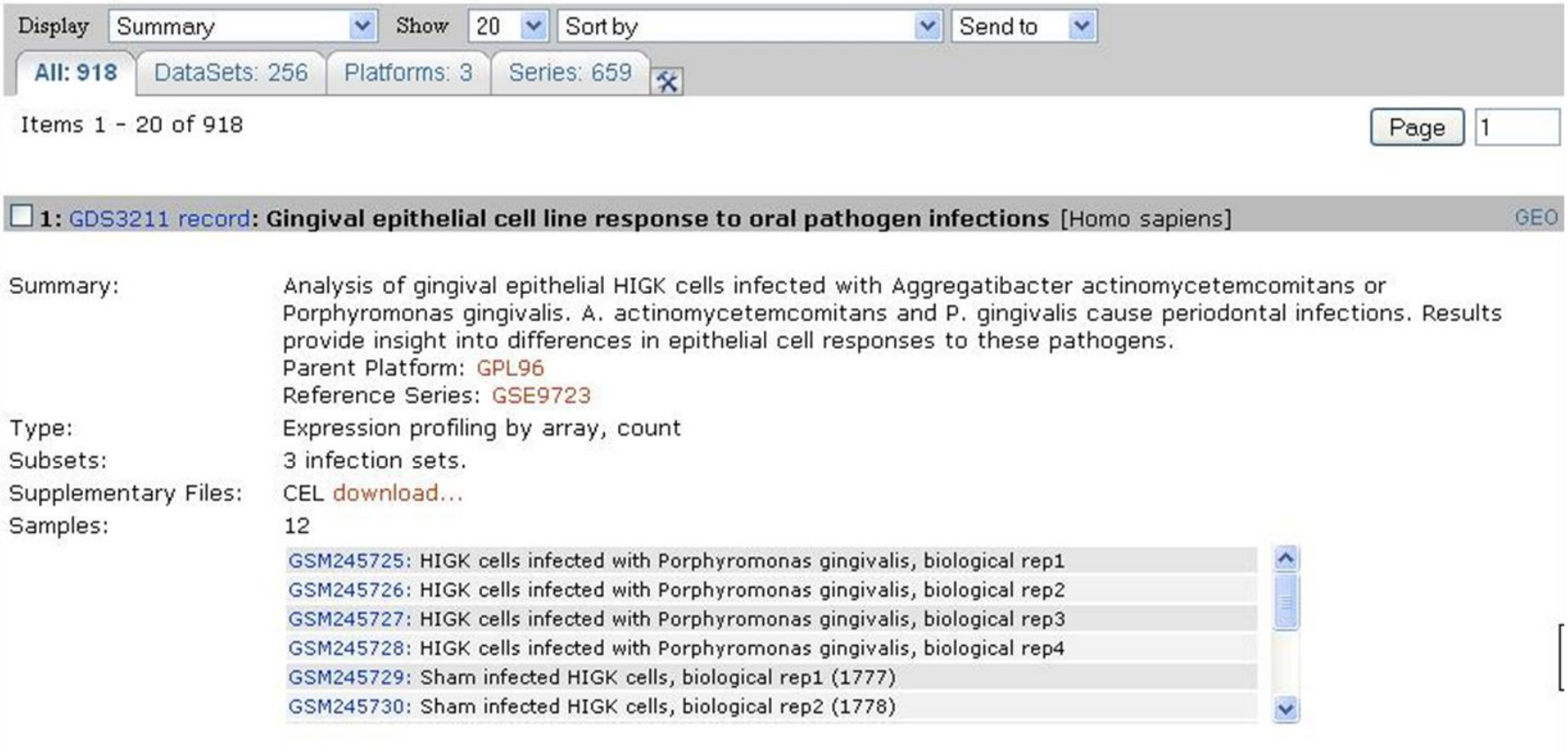


Figure 11 - Example of a snapshop view result returned by a query to NCBI GEO (Source: http://www.ncbi.nlm.nih.gov/geo/)

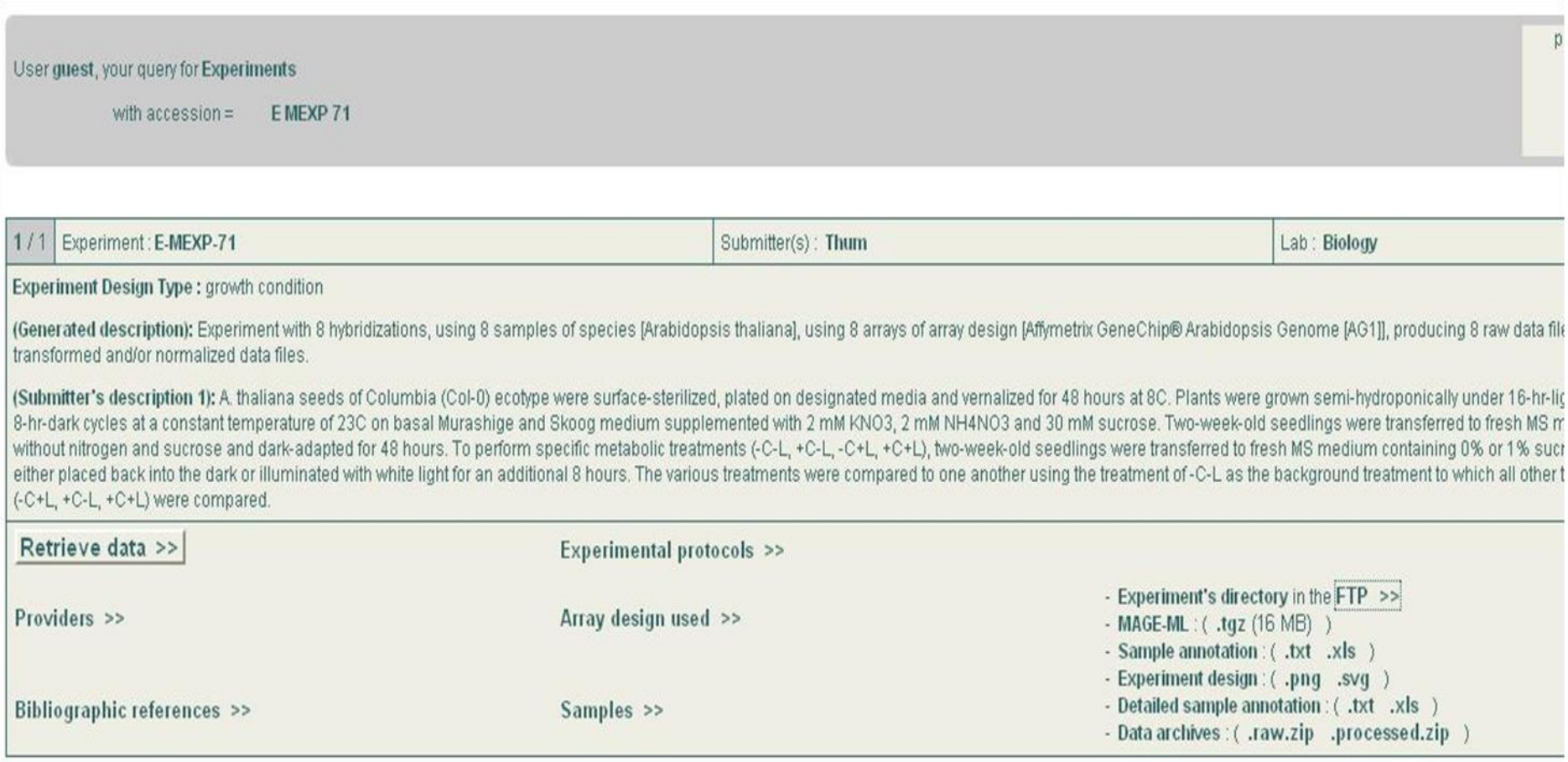


Figure 12 - Example of a snapshop view result returned by a query to EBI ArrayExpress (Source: http://www.ebi.ac.uk/arrayexpress/)

As one of the back end components of PathEx, The Data Mining Utility provides a set of algorithms to extract, parse, organize, correlate and convert relevant information: Microarray data (e.g. .CEL files) and metadata, Genes, Proteins, Pathways and Literature information. The Integration Utility manages a relational database component by loading into and updating it with appropriate structured data. The Query Handler Utility that negotiates the dataset build by checking submitted selection criteria and filters and invoking necessary sample files to build a dataset handles all user queries. PathEx, through the Updater Utility, provides a schema-evolution service that is valuable because the ongoing revision of biological data and the complexity of its database schemas imply that they are always evolving.

PathEx content (storage) logic

This component manages the PathEx data storage system: (a) the File Repositories of microarray data files: SOFT files (from GEO), MAGE-TAB files (from AE) and different biological source files used to enrich microarray characterization and (b) the Database containing structured and manually reviewed related microarray metadata and annotation information (pathway-related information, cellular components, molecular functions, disease-related information, literature information, gene-related information, protein related information, study types, etc.).

GEO SOFT files contain data tables and the accompanying descriptive information for multiple, concatenated Platforms, Samples, and/or Series records.

The integrated AE MAGE-TAB files consist of four different types of files: (a) A “raw” zip archive contains the raw data files, i.e. the files produced by the microarray image analysis software, such as CEL files for Affymetrix GeneChip, (b) The Array Design Format (ADF) tab-delimited file describes the design of an array, (c) The Investigation Description Format (IDF) tab-delimited file contains top-level information about the experiment including the title, description, submitter contact details and protocols and (d) The Sample and Data Relationship Format (SDRF) tab-delimited file containing the relationships between the samples and arrays, as well as sample properties and experimental factors, as provided by the data submitter.

Although the vast majority of these data sources are imported and integrated automatically by appropriate computing tools (API, external connectors, etc.), a significant portion of PathEx integrated data were manually recorded by a 9-team members during 3 months. Beside this data recording process, a validation process was carried out on all data integrated into PathEx database. This validation process (done by the URBM bioinformatics research team members) consisted of drawing randomly 20% of all imported and integrated data, verifying them against their original sources and making sure that the original biological meaning is preserved retaining the simpler form in the database.

PathEx navigation (browsing) logic

This component comprises a set of intuitive, interactive and easy-to-use web interfaces. They provide users with features to customize and select a dataset simply by specifying criteria not initially envisioned by those who deposited the expression array data.

PathEx design pattern

The purpose of most of biological web application/database systems is to retrieve data from a data source and display it for the researcher. Because the key flow of information is between the data source and the researcher interface, many developers might be inclined to tie these two pieces together to reduce the amount of coding and to improve application performance. However, this seemingly natural approach has some significant problems. One problem is that the user interface tends to change much more frequently than the data storage system.

Another problem with coupling the data and user interface pieces is that biological applications tend to incorporate back-end logic that goes far beyond data transmission.

The problem which rises in this case is: “How do we modularize the user interface functionality of PathEx so that we can easily modify the individual parts?”

With PathEx, we tried to consider the following facts that act on a system within this context and which were reconciled as we considered a solution to the problem stated:

- User interface logic tends to change more frequently than back-end logic, especially in Web-based applications. For example, new user interface pages may be added, or existing page layouts may be shuffled around. If presentation code and back-end logic are combined in a single object, one have to modify an object containing back-end logic every time the user interface changes. This is likely to introduce errors and require the retesting of all back-end logic after every minimal user interface change.

- In some cases, the application displays the same data in different ways. In some rich-client user interfaces, multiple views of the same data are shown at the same time. If the user changes data in one view, the system must update all other views of the data automatically.

- Designing visually appealing and efficient web interfaces generally requires a different skill set than does developing complex back-end logic (e.g. designing a database underlying a web application). Rarely does a person have both skill sets. Therefore, it is desirable to separate the development effort of these two parts to allow effective development.

- User interface activity generally consists of two parts: presentation and update. The presentation part retrieves data from a data source and formats the data for display. When the user performs an action based on the data, the update part passes control back to the back-end logic to update the data.

- In many web applications, a single page request combines the processing of the action associated with the button/link that the user clicked/selected with the rendering of the target page. Sometimes, the target page may not be directly related to the action.

- User interface code tends to be more device-dependent than back-end logic. If you want to move the application from a browser-based application to a PC-based application, you must replace much of the user interface code, whereas the back-end logic may be unaffected. A clean separation of these two parts accelerates the migration and minimizes the risk of introducing errors into the back-end logic.

- Creating automated tests for user interfaces is generally more difficult and time-consuming than creating those for back-end logic. Therefore, reducing the amount of code that is directly tied to the user interface enhances the testability of the application.

How does PathEx solve this problem?

In this research, we opted for the *Model-View-Controller (MVC)* pattern. This pattern separates the modeling of the database, the presentation, and the actions based on user input into three separate components (Burbeck, 1992):

- **Model**. The model manages the behavior and data of the application database, responds to queries from researchers, and responds to instructions to change state (usually from the controller).

- **View**. The view manages the display of query results.

- **Controller**. The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate.

Figure 13 depicts the structural relationship between the three objects.

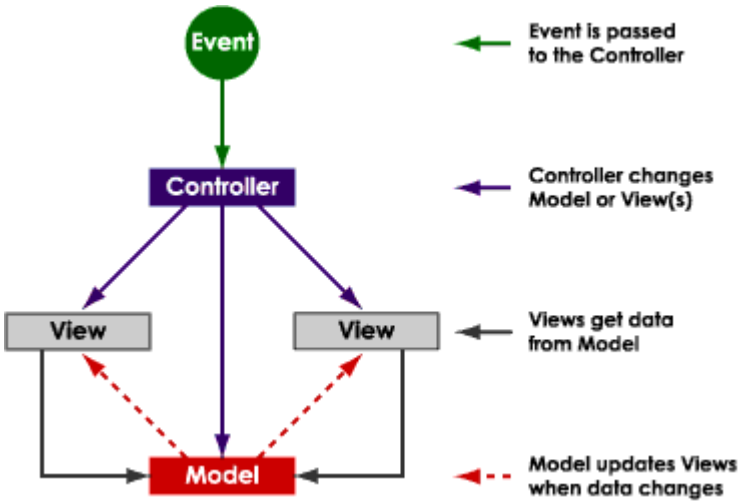


Figure 13 - MVC pattern abstraction (Source: http://www.enode.com/mvc.gif)

It is important to note that both the view and the controller depend on the model. However, the model depends on neither the view nor the controller. This is one the key benefits of the separation. This separation allows the model to be built and tested independent of the visual presentation.

In PathEx, we chose to use a passive variation of the MVC instead of an active one.

The passive variation of the MVC is employed when one controller manipulates the model exclusively. The controller modifies the model and then informs the view that the model has changed and should be refreshed (see Figure 14). The model in this scenario is completely independent of the view and the controller, which means that there is no means for the model to report changes in its state. The browser displays the view and responds to user input, but it does not detect changes in the data on the server. Only when the user explicitly requests a refresh is the server interrogated for changes.

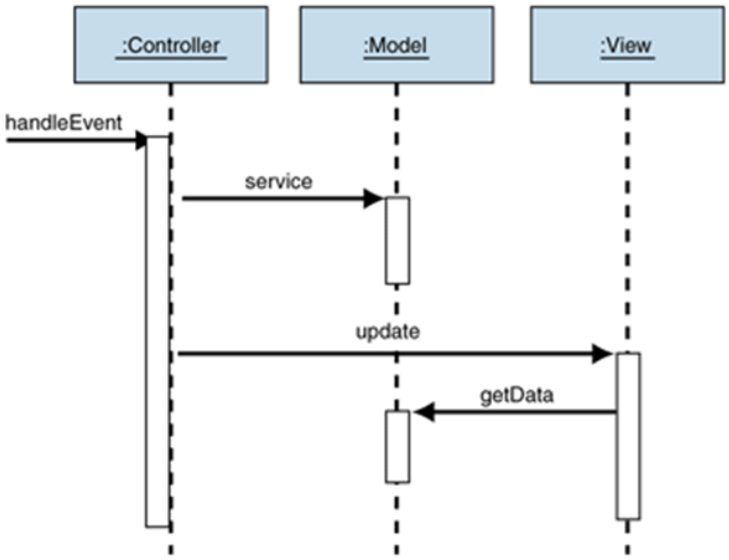


Figure 14 - Behavior of a passive MVC model (Source: http://i.msdn.microsoft.com/passive_mvc.gif)

The active model is used when the model changes state without the controller's involvement.

However, one of the motivations of using the *MVC* pattern in PathEx is to make the model independent from of the views. If the model had to notify the views of changes, we would reintroduce the dependency we were looking to avoid.

Another major advantage of using the MVC pattern is testability. Testing components becomes difficult when they are highly interdependent, especially with user interface components. These types of components often require a complex setup just to test a simple function. Worse, when an error occurs, it is hard to isolate the problem to a specific component. This is the reason why separation of concerns is such an important architectural driver. *MVC* separates the concern of storing, displaying, and updating data into three components that can be tested individually.

MVC does not eliminate the need for user interface testing, but separating the model from the presentation logic allows the model to be tested independently from the presentation and reduces the number of user interface test cases.

Benefits gained by using MVC pattern

- **Supports multiple views**. Because the view is separated from the model and there is no direct dependency from the model to the view, the user interface can display multiple views of the same data at the same time. Multiple pages in a Web application may use the same model objects. For example, currently, gViz (a co-expression networks visualization tool, developed in our lab, uses as well the same database as PathEx).

- **Accommodates change**. User interface requirements tend to change more rapidly than back-end rules. Users may prefer different colors, fonts, screen layouts, and levels of support. Because the model does not depend on the views, adding new types of views to the system generally does not affect the model. As a result, the scope of change is confined to the view.

Constraints associated to the MVC pattern

- **Complexity**. The *MVC* pattern increases the complexity of PathEx solution slightly. It also increases the event-driven nature of the user-interface code, which can become more difficult to debug (PathEx Interface grid views are complex to understand).

- **Cost of frequent updates.** Decoupling the model from the view does not mean that developers of the model can ignore the nature of the views. For example, if the model undergoes frequent changes, it could flood the views with update requests. Some views, such as graphical displays, may take some time to render. As a result, the view may fall behind update requests. Therefore, it is important to keep the view in mind when coding the model.

Implementation

Technological choices

There are many factors in consideration when you develop a web-based application. Some are hardware-related and other software-related. As the purpose of this thesis is not a comparative study on web development technologies, we limited our choice criteria on few factors. In fact, technological choices for web-based applications development are motivated by several factors such the audience, the purpose, the content and the development resources.

- **The audience.** PathEx can be used by any researcher with or without limited knowledge on microarray technologies. It was implemented with the idea of minimizing the burden related to different biological sources integrated in it.
- **The Purpose.** The primary purpose of PathEx is to generate study focused datasets, by providing to users a series of advanced query features.
- **The content.** PathEx users do not need to have any pre-knowledge of data behind it. They are just to input key criteria of their choice.
- **The development resources.** As one person-project, it was difficult to comply to all development standard requirements (source control, multiple checkouts and merge resolution) due to time constraints but we did it to ease a potential project takeover by a third party.

Technical choices

Hardware

PathEx is hosted on a dedicated cluster of 20 nodes, each having 16GB RAM. The cluster manages a storage system of up to 40TB. In case of PathEx, hardware choices were also driven by other partners' software requirements and university consolidated infrastructure support. In fact the cluster is currently integrated in a global computing cluster facility (iSCF) of the University of Namur.

Software

Apart from the availability of hardware infrastructure of the University of Namur, some factors informed the choice of the software used in the development of PathEx.

These include:

1. **Licensing.** With the exception of some of the enterprise licensing options, a software bundle brings zero licensing fees to the table. Commercial products vendors charge for the OS, database server, any specialized servers your application requires, and their IDE.

2. **Support.** Commercial products vendors will provide you with support, but again, it costs money. Free and Open Source Solutions comes with, in my opinion, a much more robust support option - the actual developers who worked on the project plus its community. Most of the time these products are free.

3. **Platform.** You can setup the stack on a wide range of platforms, by combining effectively several software solutions. Currently the most popular is the software bundle LAMP, (**L** standing for Linux (http://www.linux.org/), operating system; **A** for Apache (http://www.apache.org/), the web server; **M** for MySQL (http://www.mysql.com/), the database management system and **P** for PHP (http://www.php.net/), Perl (http://www.perl.org/) or Python (http://www.linuxfoundation.org/), the scripting language).

4. **Hardware.** This goes hand-in-hand with platform above. Linux can run very well on very inexpensive servers. It can do this because it provides administrators with the flexibility to run only what is needed to do the job. In commercial solutions, you are stuck with bloat - think how much resources needed, whereas it is typical to run headless Linux machines and use ***ssh*** to access a shell remotely.

5. **Performance.** This is tied to hardware above. When compared to a commercial product server with the same specifications; a Linux server dominates the ring in terms of performance, memory management, stability, etc...

6. **Scalability.** Scaling a web application is never cut and dry, but it is much more straight forward and cost-efficient when you're dealing with Linux-based tools. Many of today's highest-volume effective web applications are running a LAMP stack, from Facebook (http://www.facebook.com/) to Yahoo (http://www.yahoo.com/) to Google (http://www.google.com/).

Based on the above considerations and several readings, we chose the LAMP stack with RedHat-based linux OS, CentOS (www.centos.org), Apache Web server, MySQL database management system and PHP as the main scripting language . Choosing LAMP was the best way for me to gain control and power over PathEx application.

Table 3 shows a comparison table of several Operating Systems in term of license, file system handling procedures and system security considerations.

Table 3 - Comparison of major operating systems by operational license and general information

Name	Creator	First public release	Predecessor	Latest stable version	Latest release date	Preferred license
FreeBSD[12]	The FreeBSD Project	1993	386BSD	8.2	2011	BSD / Free
Linux[13]	Richard Stallman, Linus Torvalds, et al.	1992	Unix, Minix	Linux kernel 2.6.37; GNU C Library 2.11	2011	GNU GPL, GNU LGPL and other licenses / Free
Mac OS X Server[14]	Apple Inc.	2001	NeXTStep / OPENSTEP / Rhapsody, Mac OS, UNIX	10.6.4	2010	Proprietary / Commercial (APSL, GNU GPL, others)
Solaris[15]	Sun	1992	SunOS	10 10/09	2008	CDDL / Free
Windows Server (NT family)[16]	Microsoft	1993	MS-DOS, OS/2, Windows 3.x	Windows Server 2008 R2 (NT 6.1.7600)	2009	Proprietary / Commercial

[12] www.freebsd.org

[13] www.linuxfoundation.org

[14] www.apple.com

[15] www.oracle.com/us/solaris/index.html

[16] www.microsoft.com

Operating System: CentOS.

In this project, we chose CentOS[17] as the main cluster OS hosting PathEx and all other PathEx connecting applications. The main reasons being that CentOS is the most stable, secure, tested distribution for servers (Tables 3, 6,7). It is derived (almost completely, except for the logos and names) from Red Hat Enterprise Linux. RHEL goes through several rounds of quality control by Red Hat and it is the most reliable Linux distribution. CentOS is a free (no cost) derivation of RHEL.

Web server: Apache.

To our knowledge, Apache web server remains the only appropriate open source (and free) web server solution underlying many web applications. Its adaptability and customization features have made it by far the leading web server in the world when you consider stability and security (Table 4).

[17] www.centos.org

Table 4 - Comparison of different web servers by security issues and dynamic content management ability

Server	Security			Dynamic Content					
	Basic access authentication	Digest access authentication	https	CGI	FastCGI	Java Servlets	PHP	ASP.NET	IPv6
Apache[18] HTTP Server	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes, by modules	Yes
Apache Tomcat	Yes	Yes	Yes	Yes	No	Yes	Yes	No	?
IBM[19] HTTP Server	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Internet Information Services	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

[18] www.apache.org

[19] www.ibm.com

Database: MySQL

There is a wide range of database management systems (proprietary/commercial and open source/free). In this project, our database management system solution choice was constrained by budget implications, limiting our selection to open source and free solutions.

By considering only free and open source alternatives, quality of software and developer community, our choice went with MySQL[20]. MySQL is far the most flexible due to its design and conception aspects. The fact that MySQL avails several storage engines (MyISAM, InnoDB, MariaDB, ndb-cluster,…) suitable to any kind of database-based applications.

Feature set, reliability and developer community of MySQL might be an over kill for simple web applications, but any critical application we want to develop, MySQL is the choice of database. However, we recently heard that MySQL might not remain an open-source project because MySQL AB Company (owner of MySQL products) has been bought by Sun Microsystems which in turn were bought by Oracle Company. This pushes us to consider alternative but still interesting database management solutions such PostGreSQL.

Server-side Scripting: PHP

There are many equally appealing choices for scripting. Being a web application, we went with PHP[21]. Since this scripting programming language is made for web development with its version 5 Object Oriented Programming features; it has become a serious development language.

Additionally, a set of increasing open source PHP classes and functions availed by PHP developers community is another factor put in consideration while choosing PHP as the main PathEx scripting language.

JavaScript & CSS: JQuery

Among the most promising web development solutions (which can easily be integrated into and combined with PHP), JavaScript and CSS are the major ones.

[20] www.mysql.com

[21] www.php.net

Although JavaScript and CSS are not supported uniformly across browsers, it is useful to go with a framework to handle this.

For JavaScript, the undisputed leading choice is **JQuery** (http://www.jquery.com/) which has taken the web development community by storm within a short period. Today it is the JavaScript framework of choice. It supports all browsers and makes JavaScript development organized and productive.

Implementation strategies

To implement PathEx, we considered first the "System Evolution" aspect. Any bioinformatics project being subject to constant evolution due to various reasons (continuous discoveries, updates and new needs), we have made sure that PathEx update procedures are well planned and carried out without PathEx service interruption. This has been achieved by adopting the MVC pattern in designing PathEx.

The strategies followed to achieve this were to separate implementation into 3 corresponding processes: The presentation (view) process, the data (model) process and the build (control) process. Each of the three processes has several functions. The build process is somehow complex as it has three sub-processes: query sub-process, dataset builder sub-process and migration and synchronization sub-process.

PathEx Processes list

PathEx is built around a series of PHP classes, the main being PathExGrid (Table 5) and a series of JavaScript Functions (Table 8).

We took advantage of PHP Object Oriented programming features to lay down a series of reusable classes (which may be exported and exploited even in other PHP-based applications). Naming conventions of classes have been followed and a touch of appending self-explanatory names has been used, to allow any developer a simple identification of codes.

Table 5 - PathEx core class properties

PathEx PHP Grid Class	
Properties	**Description**
AjaxEnabled	Enable ajax feature of PathExGrid.
AjaxHandlePage	Get or set the service page which PathExGrid will send ajax request.
AllowFiltering	When true, all columns in grid will enable filtering feature.
AllowGrouping	When true, all columns in grid will enable grouping feature.
AllowHovering	Allow row highlighted when mouse is over.
AllowMultiSelecting	Allow selecting more than one row.
AllowResizing	When true, all resizable columns in grid will enable resizing feature.
AllowScrolling	Allow all tableview in grid scrollable.
AllowSelecting	Allow row selected
AllowSorting	When true, all sortable columns in grid will enable sorting feature.
AutoGenerateColumns	When true, grid will generate and add data-driven columns to all tableviews automatically.
AutoGenerateDeleteColumn	When true, grid will add delete command column to all tableviews automatically.
AutoGenerateEditColumn	When true, grid will add edit command column to all tableviews automatically.
AutoGenerateExpandColumn	When true, grid will add GridExpandDetailColumn to all tableviews automatically.
AutoGenerateRowSelectColumn	When true, grid will add GridRowSelectColumn to all tableviews automatically.
CharSet	Get or set the charset.
ClientSettings	Contain settings for client behaviours and messages.
ColumnAlign	Get or set alignment of column's text

ColumnValign	**Get or set vertical alignment of column's text**
ColumnWrap	**When true, all columns in grid will be in wrap mode.**
DataSource	**Get or set default datasource for all tableviews.**
DisableAutoGenerateDataFields	**Get or set the list of field names that will not be generated column by PathExGrid.**
EventHandler	**Get or set the object to handle grid events.**
FilterOptions	**Get or set default filter options for all columns in grid.**
Height	**Get or set detfault height of all tableviews.**
KeepSelectedRecords	**When true, grid will keep selected records persistent againts postback or callback.**
KeepViewStateInSession	**Make grid persistent in session.**
MasterTable	**Master table object.**
PageSize	**Get or set the default page size for tableviews.**
RowAlternative	**When true, it makes grid show alternative color for each row.**
scriptFolder	**Get or set the path to folder which contains PathExGrid script.**
ShowFooter	**When true, it makes footer of all tableviews visible.**
ShowHeader	**When true, it make header of all tableviews visible.**
ShowStatus	**When true, grid will show the status bar at bottom of grid.**
styleFolder	**Get or set the folder which PathExGrid will load css to render.**
TableLayout	**Get or set layout of all tableviews.**
Width	**Get or set detfault width of all tableviews.**

Not only are above PHP classes encompass PathEx application, but also they are the core components of the gridding approach used to develop PathEx interfaces. This approach minimizes substantially the number of standard dynamic pages which would be required to achieve the same.

As we could not list all PathEx classes, we limited our listing to the core ones.

Pathex data models

Among all data modeling tasks, modeling biological data remains one of the most difficult tasks as earlier discussed on Biological Data Challenges Section. There exist three levels of data modeling: conceptual, logical and physical. The modeling complexity increases from conceptual to logical to physical. When modeling PathEx database, we started with the conceptual data model. This enable us to understand at high level what the different entities in our data were and how they related to one another. Then we moved on to the logical data model to understand the details of our data without worrying about how they would be implemented. Finally we ended with the physical data model which led to knowledge about the exact implementation of our data model in MySQL. In PathEx project, we noticed that the conceptual data model and the logical data model can be considered as a single deliverable due to the way it integrates various data sources.

Conceptual modeling

The following illustration (Figure 15) purely documents the data and information within the whole PathEx system and how it is used. It identifies the highest-level relationships between the different PathEx database component entities. At this level of modeling no attribute was specified and primary keys were specified.

Conceptual Diagram

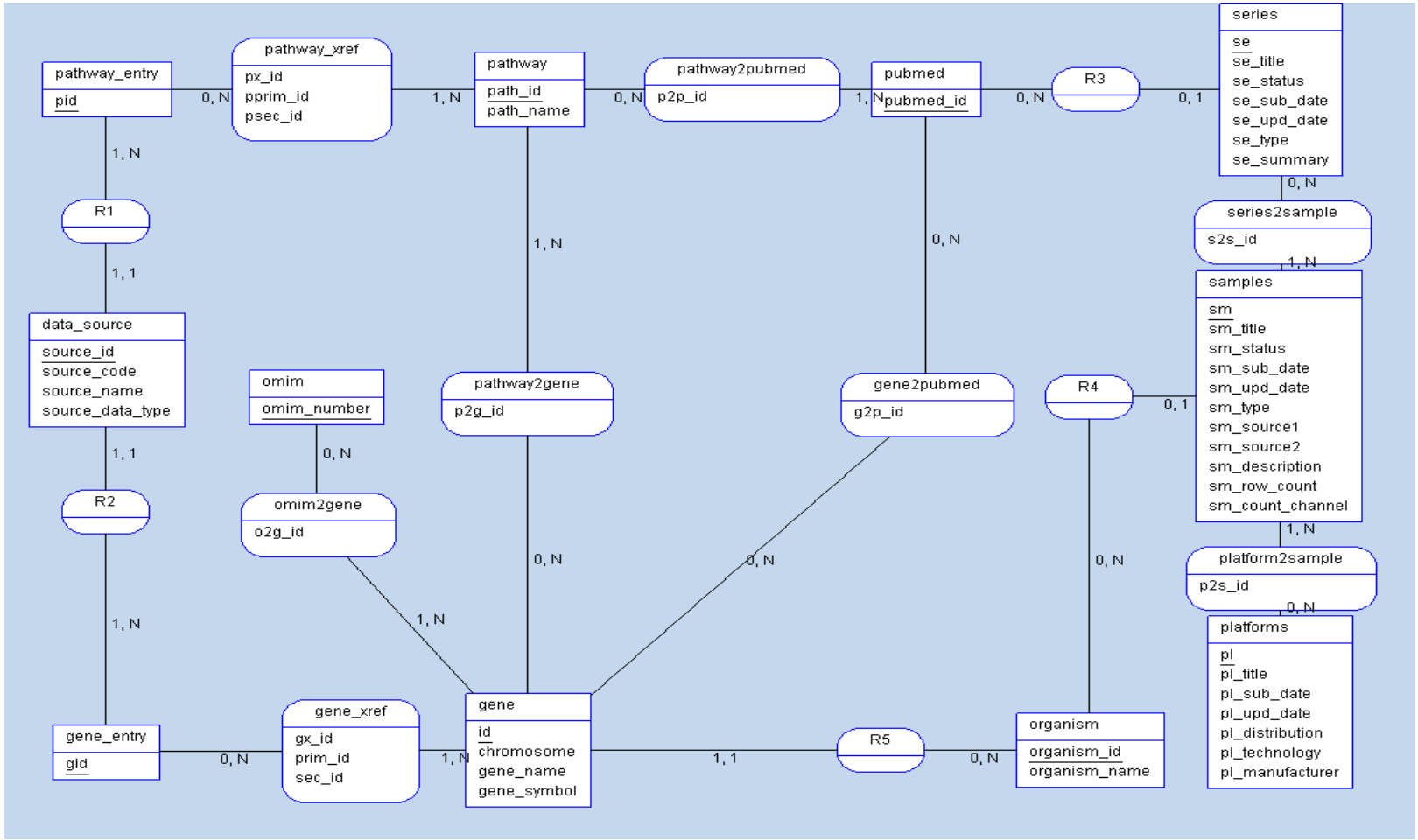


Figure 15 – PathEx Conceptual Data Model

This model identifies the highest-level relationships between the different PathEx entities.

Inherent Enhanced Entity Relational (EER) diagram

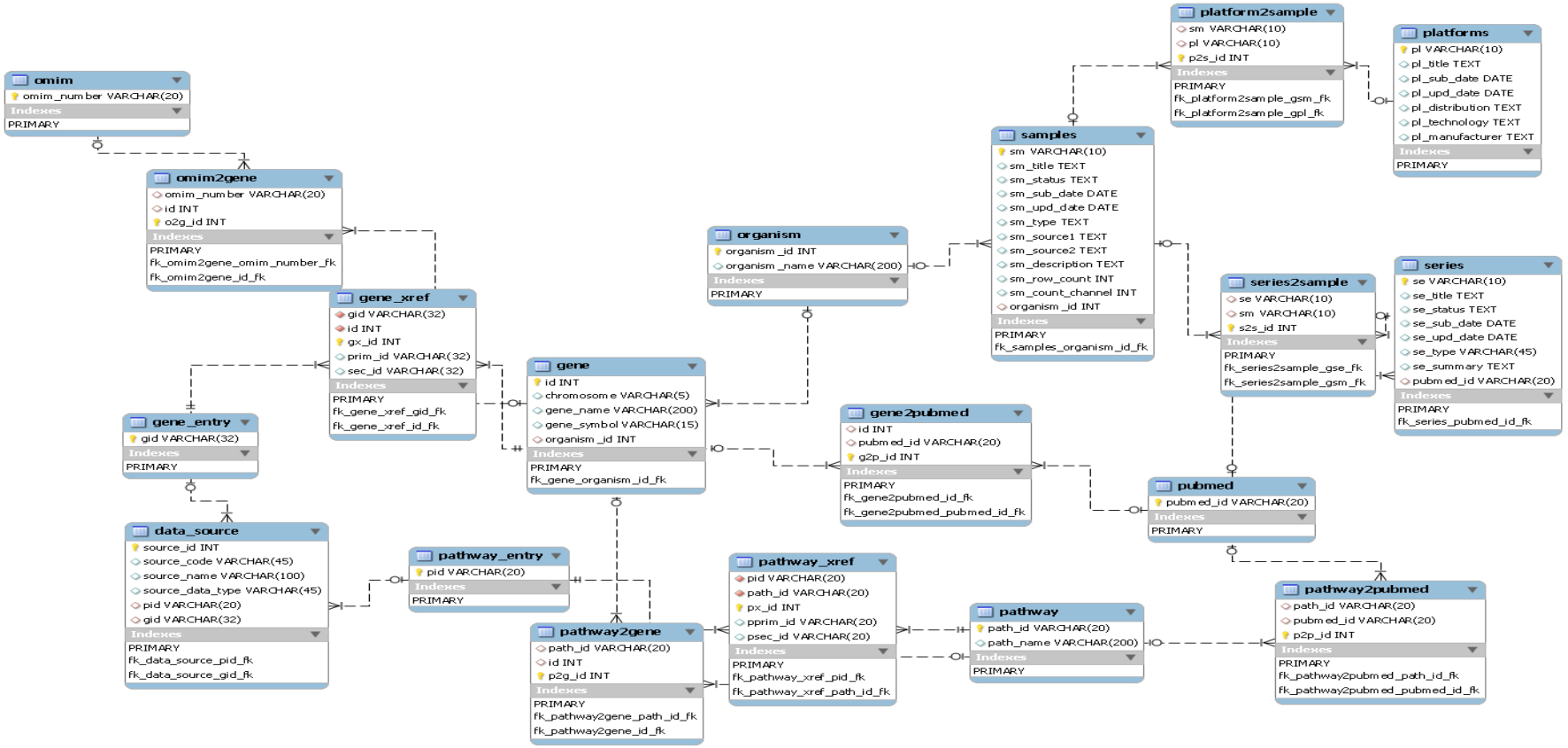


Figure 16 – PathEx Enhanced Entity Relation model

The PathEx conceptual model (earlier discussed) depicted high-level relationship and had to be translated into an inherent Entity Relation (ER) model. However, we came to notice that a simpler ER model diagram does not reflect all key elements from the conceptual model diagram. We ended up by proposing a new type of ER model diagram, namely the Enhanced ER model diagram (Figure 16)

Logical modeling

The advantage of laying out of this type of modeling on PathEx, is that it focused to the aim of creating a data model that is completely independent from any particular DBMS or software/hardware platform and is directed derived from the PathEx conceptual model.

This model describes the PathEx data in as much detail as possible, without regard to how they will be physically implemented in the database. Features of a logical data model include:

- All PathEx entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is specified.
- Foreign keys (keys identifying the relationship between different entities) are specified.
- Normalization occurs at this level.

There are different logical model notations. Even if we modeled PathEx by ER-Merise method, we thought it might be interesting to show the Baker logical model notation as well (Figure 17).

PathEx logical diagram

Baker Model Option

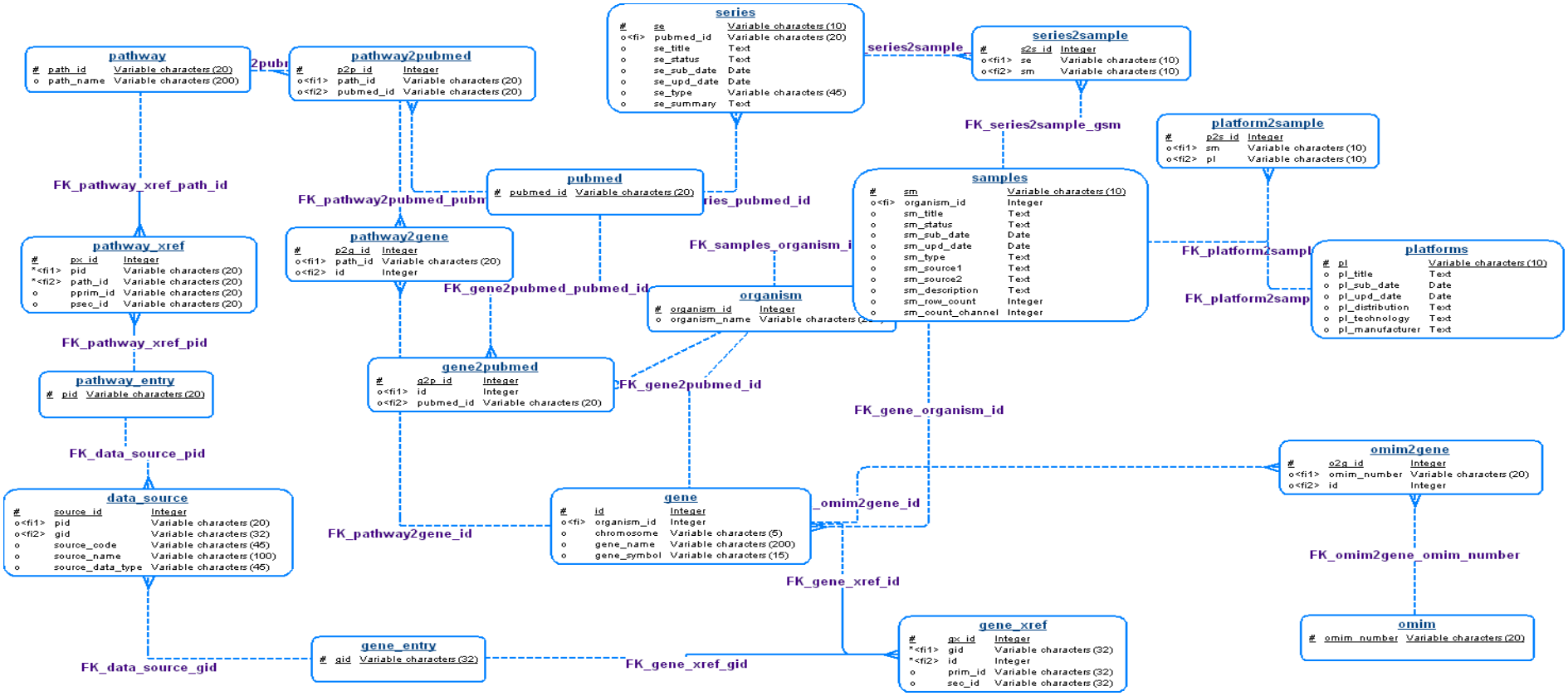


Figure 17 – PathEx Logical Data Model (Baker notation)

This model describes the PathEx data in as much detail as possible, without regard to how they will be physically implemented.

```
gene_entry (gid)

gene (id, chromosome, gene_name, gene_symbol, #organism_id)

pathway_entry (pid)

pathway (path_id, path_name)

organism (organism_id, organism_name)

data_source (source_id, source_code, source_name, source_data_type, #pid, #gid)

samples (sm, sm_title, sm_status, sm_sub_date, sm_upd_date, sm_type, sm_source1,
sm_source2, sm_description, sm_row_count, sm_count_channel, #organism_id)

pubmed (pubmed_id)

series (se, se_title, se_status, se_sub_date, se_upd_date, se_type, se_summary, #pubmed_id)

platforms (pl, pl_title, pl_sub_date, pl_upd_date, pl_distribution, pl_technology,
pl_manufacturer)

omim (omim_number)

pathway2gene (path_id, id, p2g_id)

gene_xref (gid, id, gx_id, prim_id, sec_id)

pathway_xref (pid, path_id, px_id, pprim_id, psec_id)
```

This model lists out all PathEx tables and their respective fields. Some developers prefer this kind of model in place of graphical ones

Physical Modeling

This model (Figure 18) represents how the model is built in the PathEx database. It shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables in MySQL.

Its main features include:

- Specification all PathEx tables and columns.
- Foreign keys are used to identify relationships between PathEx tables.
- Denormalization.

On this step, we turned previous PathEx models into DBMS (MySQL) instructions. At this level, since all other models have been well planned and analyzed, the remaining tasks were to:

- Convert PathEx database component entities into MySQL tables.
- Convert PathEx database component relationships into foreign keys.
- Convert PathEx database component attributes into columns.
- Modify the PDM based on physical constraints / requirements (e.g. MySQL storage engines: MyISAM, InnoDB, MariaDB…).

PathEx Physical Diagram

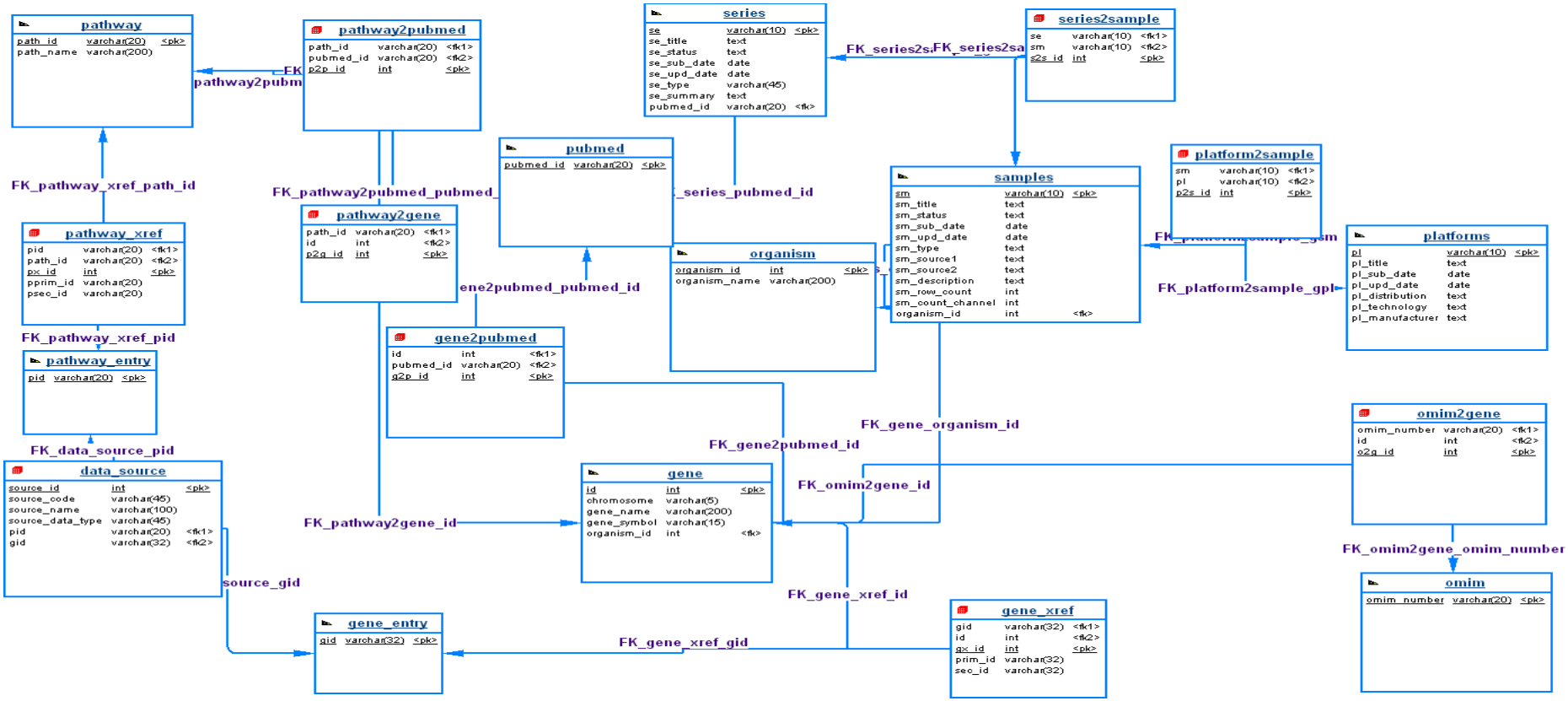


Figure 18 – PathEx Physical Data Model

This model represents how the PathEx model will be built in the MySQL database. It shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.

Pathex coding & features

User interfaces coding

PathEx web interfaces were coded using rich-client API, with many features topped with an unbeatable performance core API named PathExGrid. This API contains built-in explorer-like navigator and several advanced features.

```
$gse = new GridTableView();
$gse->DataKeyNames = "ID";
$gse->ShowGroupPanel = true;
$gse->GroupPanel->ItemConnector = ">";
$gse->Width = "98%";
$gse->Height = "300px";
$gse->DataSource = $ds_experiments;
$gse->AddRelationField("ArrayID","ArrayID");
$gse->AutoGenerateColumns = true;
$gse->AutoGenerateRowSelectColumn = true;
$gse->AllowSelecting = true;
$gse->AllowFiltering = true;
$gse->AllowSorting = true;
$gse->AllowGrouping = true;
$gse->AllowScrolling = true;
$gse->AllowMultiSelecting = true;
$gse->KeepSelectedRecords = true;
$gse->ColumnWrap = true;
$gse->DisableAutoGenerateDataFields = "ID,ArrayID";
$gpl->MasterTable->DataSource = $ds_platform;
$gpl->MasterTable->DataKeyNames = "ArrayID";
$gpl->MasterTable->AutoGenerateExpandColumn = true;
$gpl->MasterTable->AutoGenerateColumns = true;
$gpl->AllowFiltering = true;
$gpl->AllowSorting = true;
$gpl->ColumnWrap = true;
$gpl->MasterTable->AddDetailTable($gse);
$gpl->MasterTable->Pager = new GridPrevNextAndNumericPager();
$gpl->MasterTable->Pager->PageSize = 5;
$gpl->MasterTable->Pager->PageSizeOptions = "5,10,20,40,50";
$gpl->MasterTable->Pager->PageSizeText = "Change Page Size : ";
$gpl->Process();
```

Figure 19 - PathEx grid class features.

PathExGrid key features (e.g. Figure 19):

- Very easy to use,
- Rich client-side API and events,
- AJAX Capability,
- State persistence through postback and ajax callback,
- Scrolling with position persistence,
- Sorting, Filtering, Grouping, Row (Multi)Selecting (Figure 20)
- Full (De)Selecting (Figure 21)

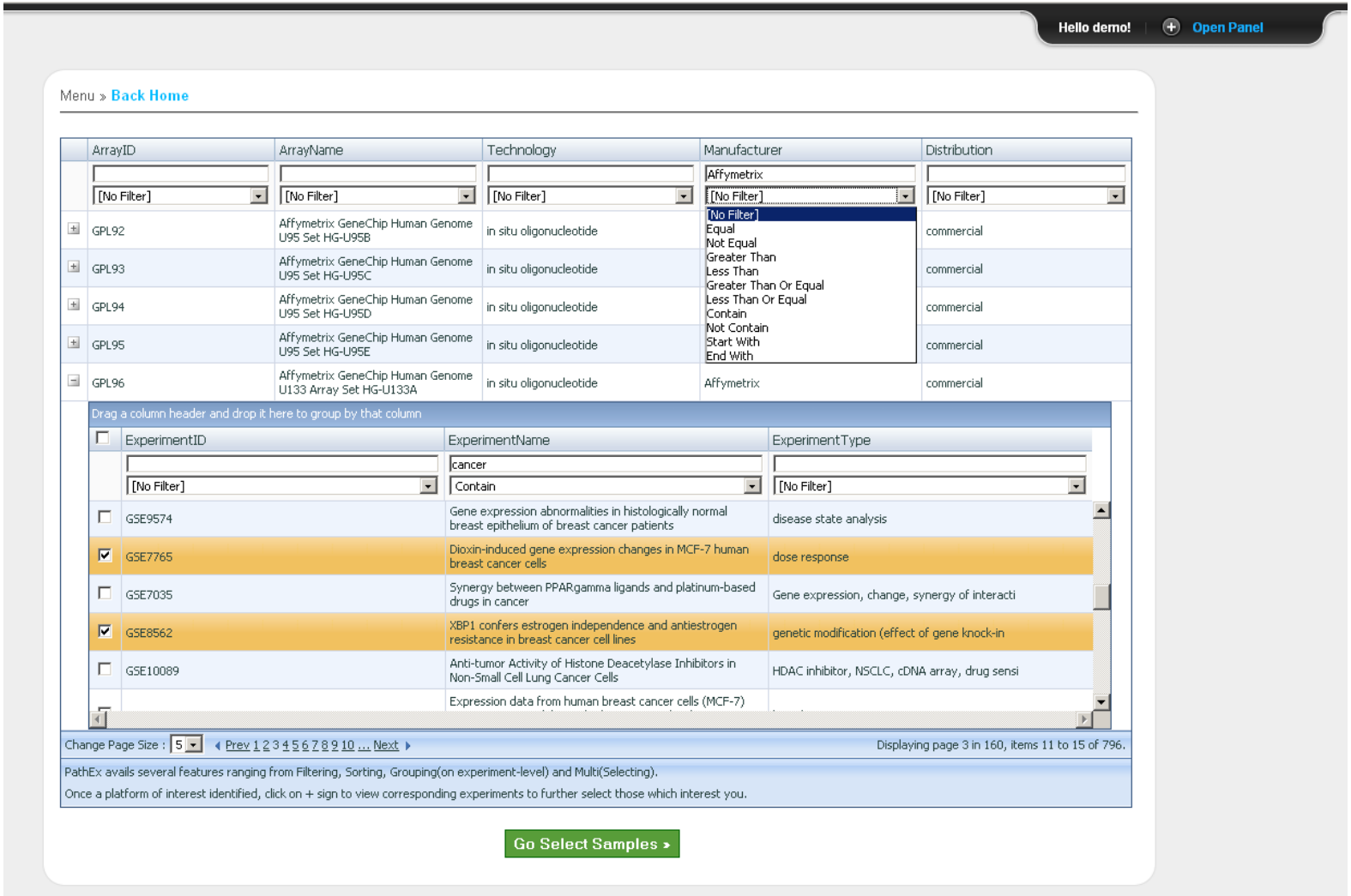


Figure 20 - PathEx sorting, filtering, grouping, row multi (selecting) features

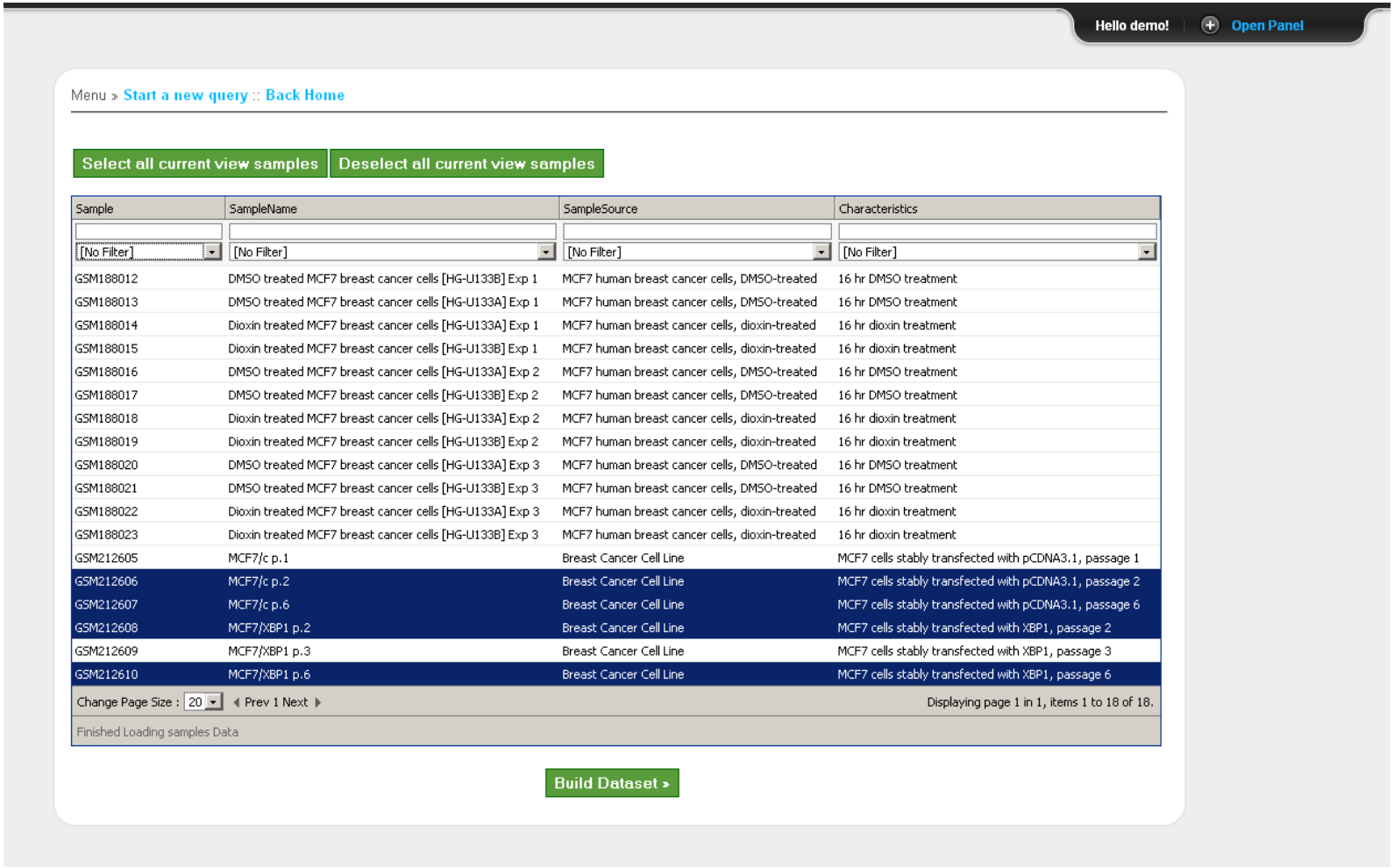


Figure 21 - PathEx full (de)select features

PathEx contains a full-featured dataset builder and experiments/samples search and browse tool. Users can browse expression data (platforms, experiments and samples), set the viewing options, perform advanced searches, select platforms/experiments/samples of interest, set the dataset(s) building options, build dataset(s) and download new generated dataset(s).

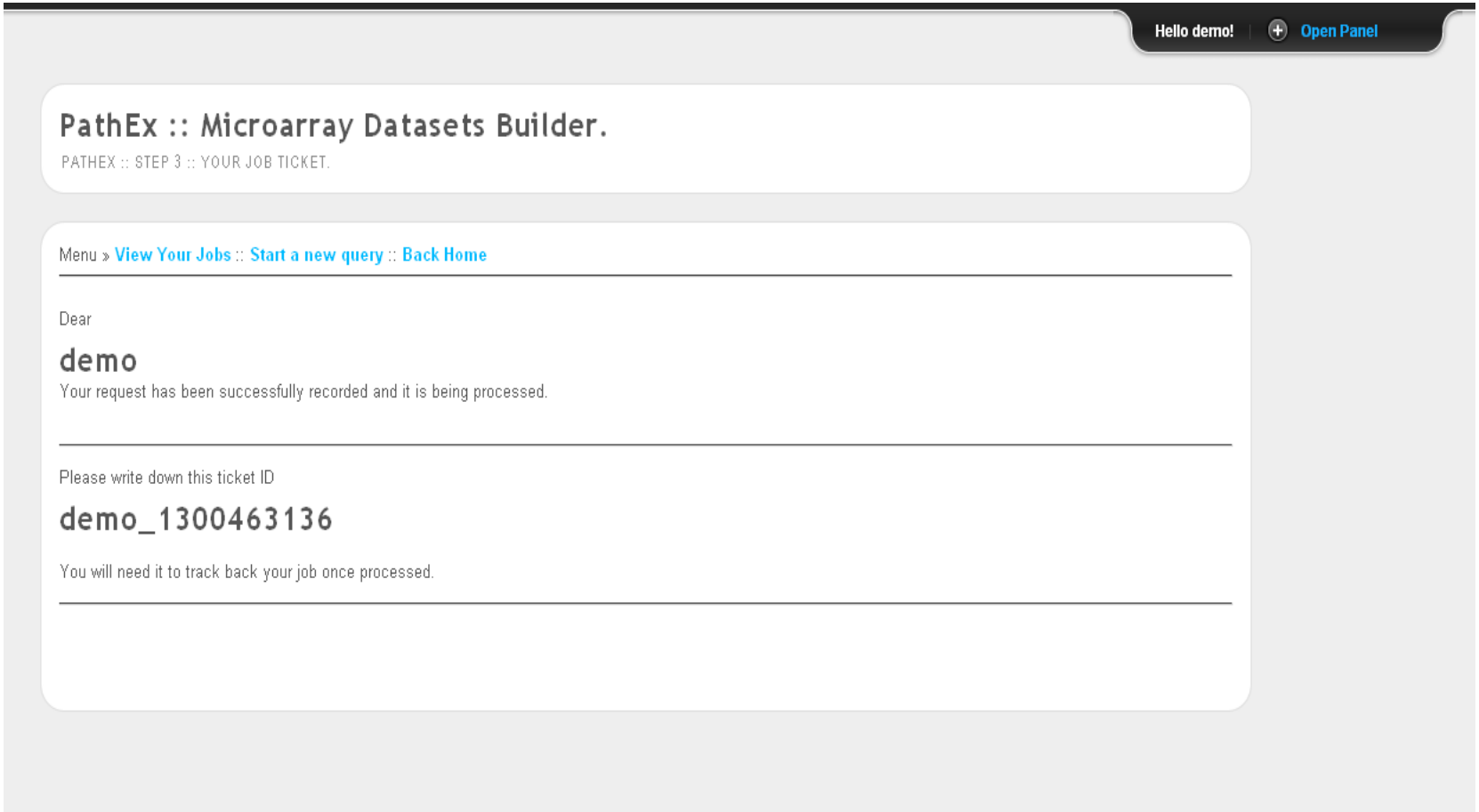


Figure 22 - PathEx datasets builder ticketing system

PathEx presents online directly. Each job request is associated to his/her owner and is assigned a ticket (Figure 22) identification code. The resulting files are compressed in one ZIP file (Figure 23) that can be retrieved any time for further use, each user having its own dedicated work environment.

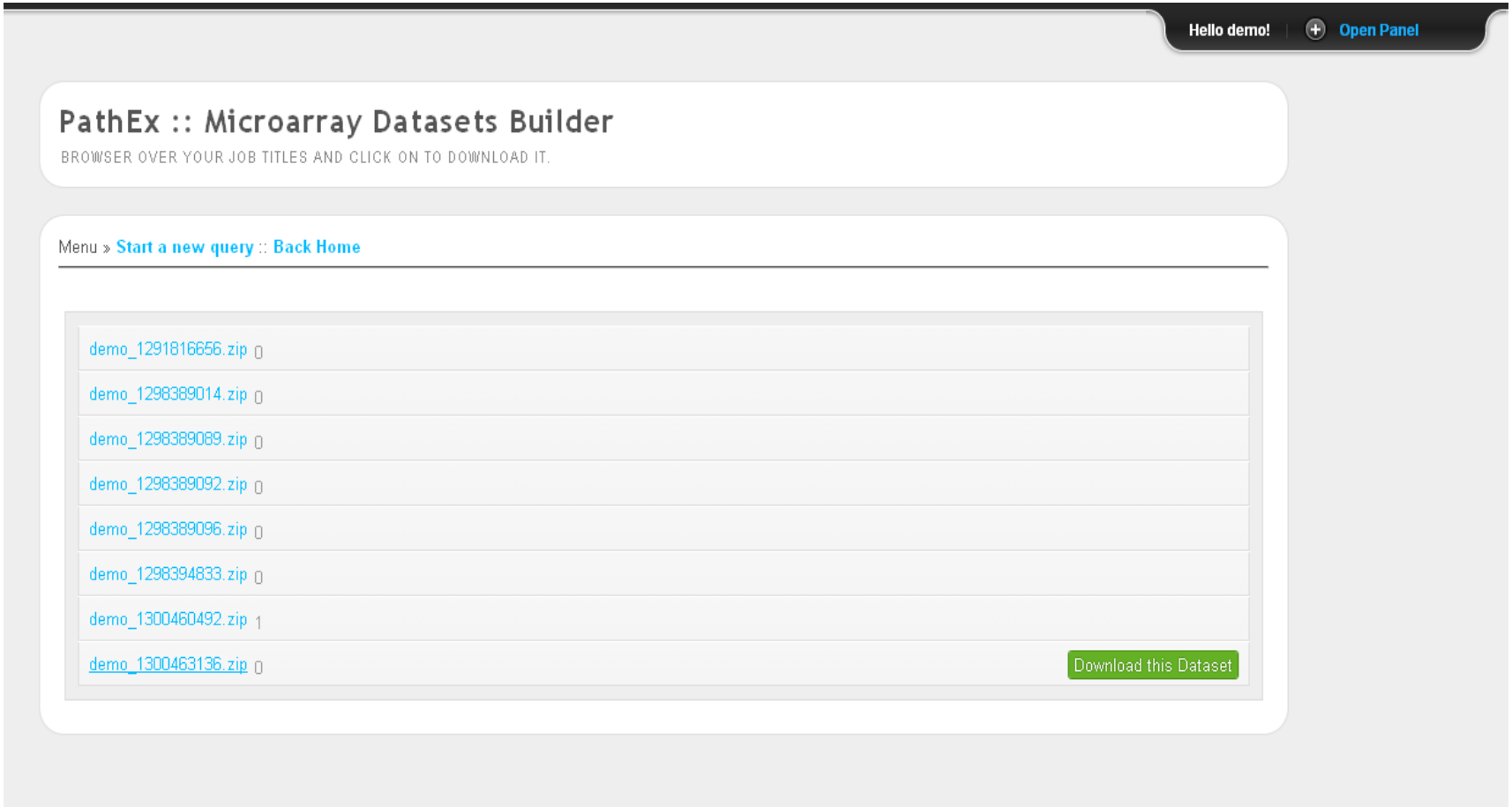


Figure 23 - PathEx dataset cart interface

On demand, an advanced job interface can be availed. This interface helps those who may wish to organize their datasets environment be creating new folders and files, merge, copy, rename or delete existing folders and files (Figure 24). We did not avail this interface to everyone due to storage issue.

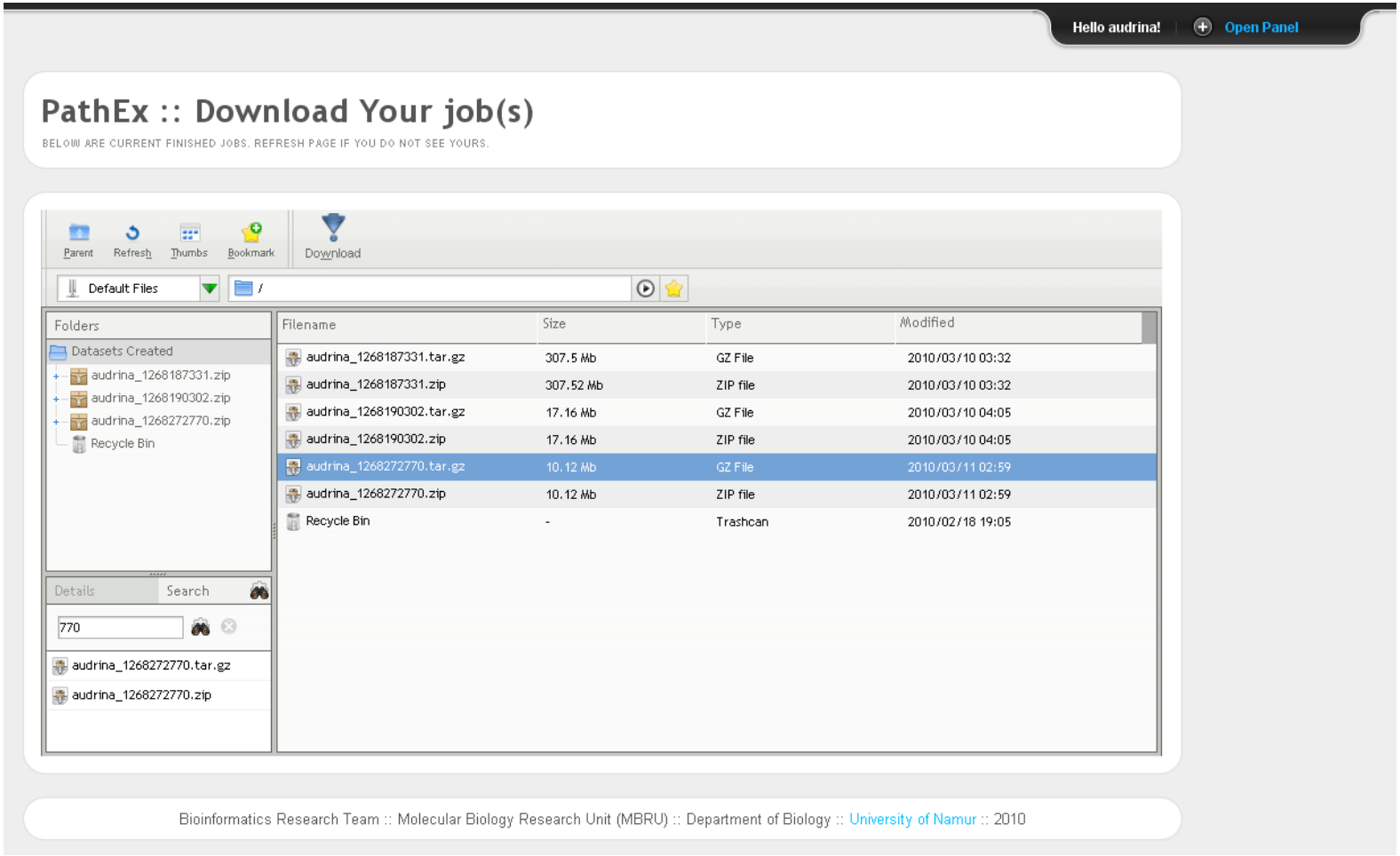


Figure 24 - PathEx advanced dataset cart interface

Back-end coding

PathEx back end utilities were coded using Perl, Java and Bash. The migration and synchronization tools were all coded in Perl while the dataset building tools were coded in Bash.

The following (Figure 25) is an example of back-end snapshot Java codes.

```
public class PathEx extends Thread {
		private String PathExQuery, PathExResults, PathExSamples, PathExErrors;
		private File queryDir;
		private Database db;
		private String smtp, adminMail;

		public PathEx() throws Exception {
				File configFile = new File("path.cfg");
				FileReader fr = new FileReader(configFile);
				BufferedReader br = new BufferedReader(fr);
				String line;
				while ((line=br.readLine()) != null){
						if (line.startsWith("query")){
								PathExQuery = line.split("=")[1].replace('\\', '/');
						}else if (line.startsWith("samples")){
								PathExSamples = line.split("=")[1].replace('\\', '/');
						}else if (line.startsWith("results")){
								PathExResults = line.split("=")[1].replace('\\', '/');
						}else if (line.startsWith("errors")){
								PathExErrors = line.split("=")[1].replace('\\', '/');
						}else if (line.startsWith("smtp")){
								smtp = line.split("=")[1].replace('\\', '/');
						}else if (line.startsWith("admin")){
								adminMail = line.split("=")[1].replace('\\', '/');
						}else{
								System.err.println("Unknown parameter in path.cfg : " + line +
												"\nYou should use 'query', 'samples',
'results', 'errors', 'admin' or 'smtp' parameters.");
						}
				}
				if (PathExQuery == null) PathExQuery = "PathExQuery";
				if (PathExResults == null) PathExResults = "PathExResults";
				if (PathExSamples == null) PathExSamples = "PathExSamples";
				if (PathExErrors == null) PathExErrors = "PathExErrors";
				if (smtp == null) smtp = "smtp.fundp.ac.be";
				if (adminMail == null) adminMail = "pathexdb+error@gmail.com";
				queryDir = new File(PathExQuery);
				File dbParam = new File("database.cfg");
				fr = new FileReader(dbParam);
				br = new BufferedReader(fr);
				String[] params = new String[5];
				int i=0;
				while ((line=br.readLine()) != null){
						params[i] = line.replace("address:", "").replace("schema:", "").replace("port:",
"").replace("login:", "").replace("pass:", "").trim();
						i++;
				}
				db = new Database(params[0], params[1], params[2], params[3], params[4]);
		}
}
```

Figure 25 - Snapshot of datasets builder system Java codes

PathEx Update Issues

Most of the biological sources integrated in PathEx keep on changing their data structures (models and format) and query interfaces (web, API…). Equally, some information used to annotate microarray metadata contained in PathEx were collected, reviewed and encoded manually.

This aspect has hampered automation of PathEx update procedures and has made almost mandatory to have a staff keeping on adapting migration and synchronization update scripts.

The fact that PathEx was developed also as a thesis project does not simplify the evolution and sustainability of this web solution.

In this context, we developed a parallel update system allowing any non-bioinformatician researcher (but with little pre-knowledge in bioinformatics) to encode manually updates and let PathEx take over remain update tasks.

However, we wish to emphasis that this can only be a temporary solution as it is better to have a full automated solution as PathEx was intended in its initial status.

Chapter IV

Results & Application on real cases

Results

Successful functional annotation of microarray data can lead to information useful for the elucidation of molecular mechanisms underlying various biological phenomena. It is therefore important to improve on the analysis of experimental gene expression data.

Microarrays are the major source of data for gene expression levels, allowing the expression of thousands of genes to be measured simultaneously. There are now many publicly available microarray datasets under different experimental conditions. However, a key issue with microarray datasets is often referred to as the curse of dimensionality. A single microarray dataset usually contains a large number of genes (hundreds or thousands) but the number of observations is much lower - generally tens or up to a few hundred at the very most. This makes it very difficult to extract reliable biological information from a single dataset.

Our research work addressed this need, enhancing data analysis performance in the functional annotation of genes by combining multiple microarray studies, using data resources that are increasingly common to scientists.

It is not uncommon that experiments on identical or similar sets of genes are conducted by multiple laboratories for various functional studies of these genes. Much of such data are often available to researchers for their data analysis, either through collaborators or from online gene expression databases. It will be useful to combine data from different microarray studies to improve the microarray data analysis results.

Integration of datasets increases the sample size and improves the analytical accuracy and statistical power of the test.

However, blindly combining all available datasets may not always improve the analysis results. It is important to be selective of the datasets for inclusion. One of the most pressing challenges in the field of microarray technology is how to integrate results from different microarray experiments or combine data sets prior to the specific analysis. Proposed method is simple but useful to combine several data sets from different experimental conditions. With this method, biologically useful information can be detectable by applying various analytic methods to the combined datasets with increased sample size.

An inescapable problem with combining several microarray data sets is the variation of expressions between data sets. In cases where the microarray analyses are from different experimental conditions, integration without transformation may skew the expression ratios of the same genes from different data sets. When the experimental bias exceeds biological variation, the use of microarray data sets without adjustments for this bias may make biological variation unidentifiable, meaning that reliable results cannot be obtained. In addition, due to the limited numbers of available microarray experiments, the motivation to use the whole dataset, regardless of platforms or experimental procedure, is increasing.

In this work, we showed our intuition that combining data from multiple experimental studies can improve microarray statistical data analysis results is correct, even in the cases where the scientific focus and experimental conditions of the individual microarray studies differ from one another.

We know that to conduct a routine microarray study analysis, we need (a) a dataset of interest, (b) an appropriate analysis method and (c) a means to evaluate, interpret and validate the results obtained. Currently, benchmarking studies have often emphasized the importance of selection of the analysis methods. This agrees with our recent benchmarking analysis, where we showed that the choice of appropriate analysis methods is crucial for the accuracy of the expected results. Recently, a re-analysis conducted on Golden Spike data by Pearson (Pearson, 2008) outlined the characteristics of an ideal dataset: (a) a realistic spike-in concentration, (b) a mixture of up- and down-regulated genes, (c) unrelated fold change and intensity and (d) *a large number of arrays*. Based on these criteria, we believe that custom selection of a dataset to analyze is crucial.

As the principal objective of a microarray analysis is to reduce variability, we should consider unexploited ways to do this, particularly in light of the outcome of several studies (De Hertogh, et al.) that postulated a complex relationship between variability and expression level. We think that, without minimizing other sources, variability can be reduced by intelligently selecting a focused dataset (e.g. dataset related to a specific pathway, pathology, organ and other factors)

However, as there are no existing tools to automatically select such a dataset, PathEx constitutes an important tool in this context.

With its enriched content and advanced selection features, PathEx provides simple and easy-to-use interfaces (Figures 21, 22, 23, 24, 26) to help users avoid the burden of thinking about complex queries. It combines flexibility, fast processing, accuracy and an easy-to-understand search system using biological tag criteria.

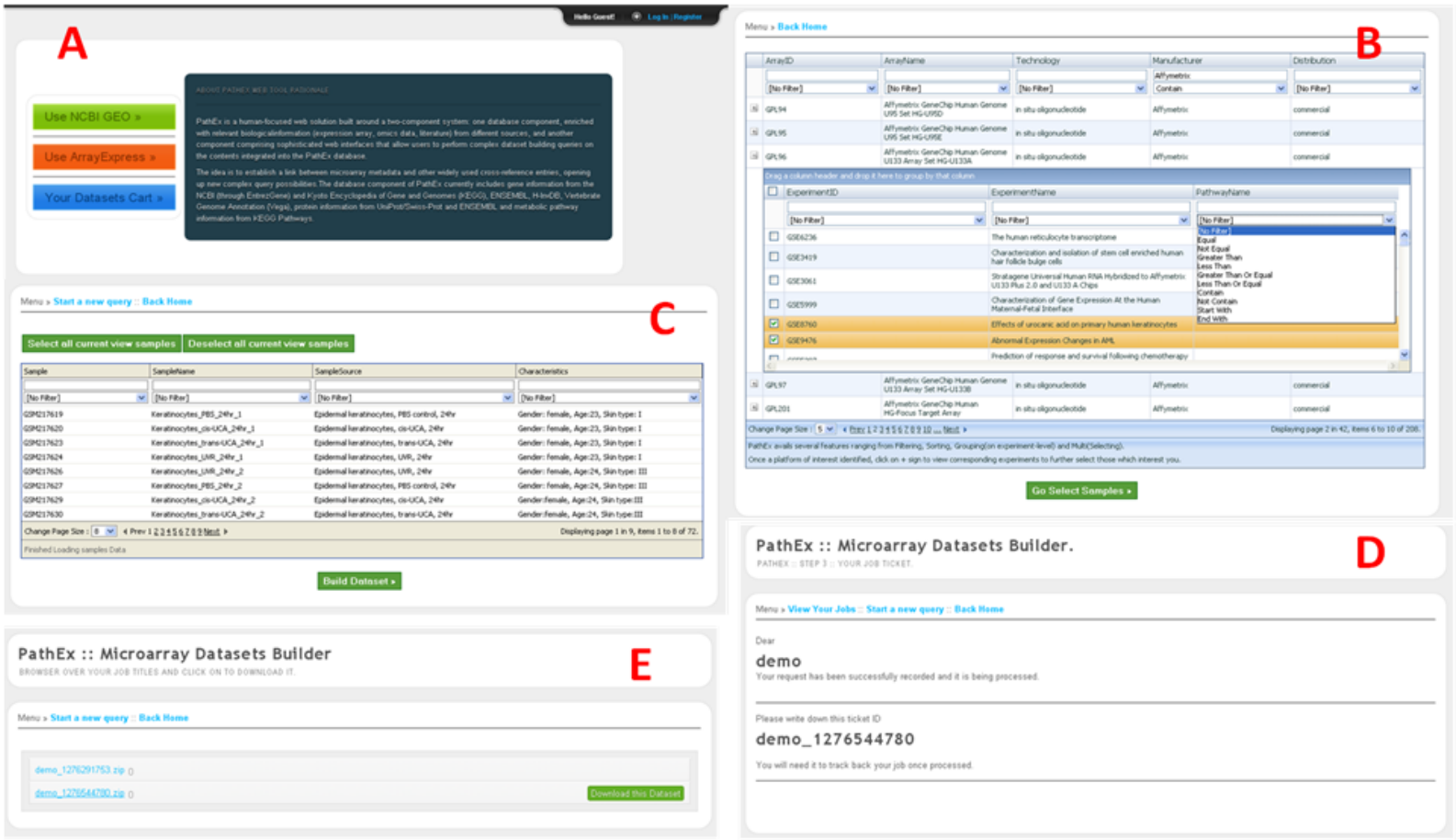


Figure 26 – (A) the entrance page after login, (B) the query interface to (Multi) select experiments of interest, (C) the dataset-to-include samples setting interface (D) the dataset-to-create ticket code and (E) the user-driven dataset cart.

With its multiple level consecutive query interfaces, PathEx maximizes the user searching process and keeps users informed of each query task filter output at each level of dataset selection within an interactive grid.

A user is provided with a specific area and interfaces according to settings chosen on the entrance page.

PathEx provides three-level selection interfaces, related consecutively on the organizational levels of the microarray data (platforms, studies and samples). Besides a search area, coupled with a set of filters ("equals", "contain", "does not contain" and others) at each level to allow for criteria-driven selection of datasets, there are advanced features to ease selection such as grouping, sorting and multiselecting.

Through the navigational settings, the user specifies the kind of keywords to query PathEx, to allow PathEx to display a customized interface. This approach was chosen to ease dataset selection and present clear interfaces.

Many keyword types can be used to query PathEx (e.g. Accessions: gene IDs, gene symbols, protein IDs, OMIM number, and PubMed IDs; Factors: Metabolic pathway names, pathology names, tissues, and organ and experiment types).

For each dataset selection request, a user is given a building ticket to trace the job process and download it when finished. The outcome is a compressed file containing all samples files related to the criteria submitted.

There are two ways of retrieving the datasets generated. Any user may retrieve its own datasets through the job cart, as it is name-driven. To evaluate the performance of PathEx, we tested it by selecting a customized dataset related to *"lung cancer"* from *"all"* *"GEO"* experiments of the type *"Affymetrix"* GeneChip *"HG-U133A"*. By submitting the five highlighted search keys to PathEx and applying appropriate filters, we ended up, in less than 30 seconds, with a dataset of 108 samples. By cross-checking we found that no any samples are related to lung cancer. Fact which confirms that PathEx was able to find expected samples 100% accurately.

Application on real cases

Due to rapid advances of microarray data analysis technologies and specialized foci of previous microarray researchers, it will be of significance to re-analyze some published microarray data. In addition, with PathEx, it becomes possible to study some new research projects using deposited microarray data from different laboratories.

We validated the PathEx efficiency by applying it to a real known biological phenomenon case, and further tested its retrieval ability by retrieving previously published expression data from two recently published studies.

Case Study: Meta-analysis on Genes regulated by Hypoxia and involved in a metastatic phenotype in cancer cells.

In a recently published work (Pierre, et al., 2010) (in which the author of this thesis contributed), we tried to evaluate the effectiveness of PathEx. We used it to try to find genes involved in the metastasis of cancer cells induced by hypoxia. Though many advances have been made in this field, all of the mechanisms involved are still not well understood. It is known that the expression of specific genes is modified in primary tumor cells to detach, migrate and invade surrounding tissues. But the integration of all of the associated data is a problem.

We approached the principal investigator of this research, and we asked him to provide us with keywords he thought were matching his datasets. He provided us with the following keywords: ***metastasis***, ***hypoxia***, ***prostate*** ***and*** ***blastoma***. We worked together to search and build a new dataset using the keywords he provided with. Knowing that he was interested mainly by Affymetrix GeneChip, we started by selecting platform by using keywords Affymetrix GeneChip, then we processed queries further.

In the first phase of meta-analysis, we used PathEx to select datasets about ***metastasis***, ***hypoxia***, ***prostate*** ***and*** ***blastoma***. Out of the seventeen (17) manually (Table 14) retrieved datasets (during three months), thirteen (13) were retrieved in less than five (5) minutes and four (4) were not retrieved at all. As some of the selected datasets were not available in GEO or AE, we investigated why PathEx could not find them. We came to notice that three (3) out of four (4) not retrieved were proprietary datasets (not available for the public directly, and PathEx does not index proprietary experiment datasets). The remaining dataset does not have any direct or indirect relation with the provided keywords. By trying to understand why, we noticed that during the manual review and curation processes, wrong information were recorded by errors. However this has been corrected as a second proof-review was conducted by different bioinformatics 'team researchers.

After retrieving the above mentioned datasets, we decided to re-analyze them using the same methods as the one used in the original work. The surprise comes by noticing that results did not change.

Below (Figure 27, 28 and 29) are some snapshots of the datasets building processes.

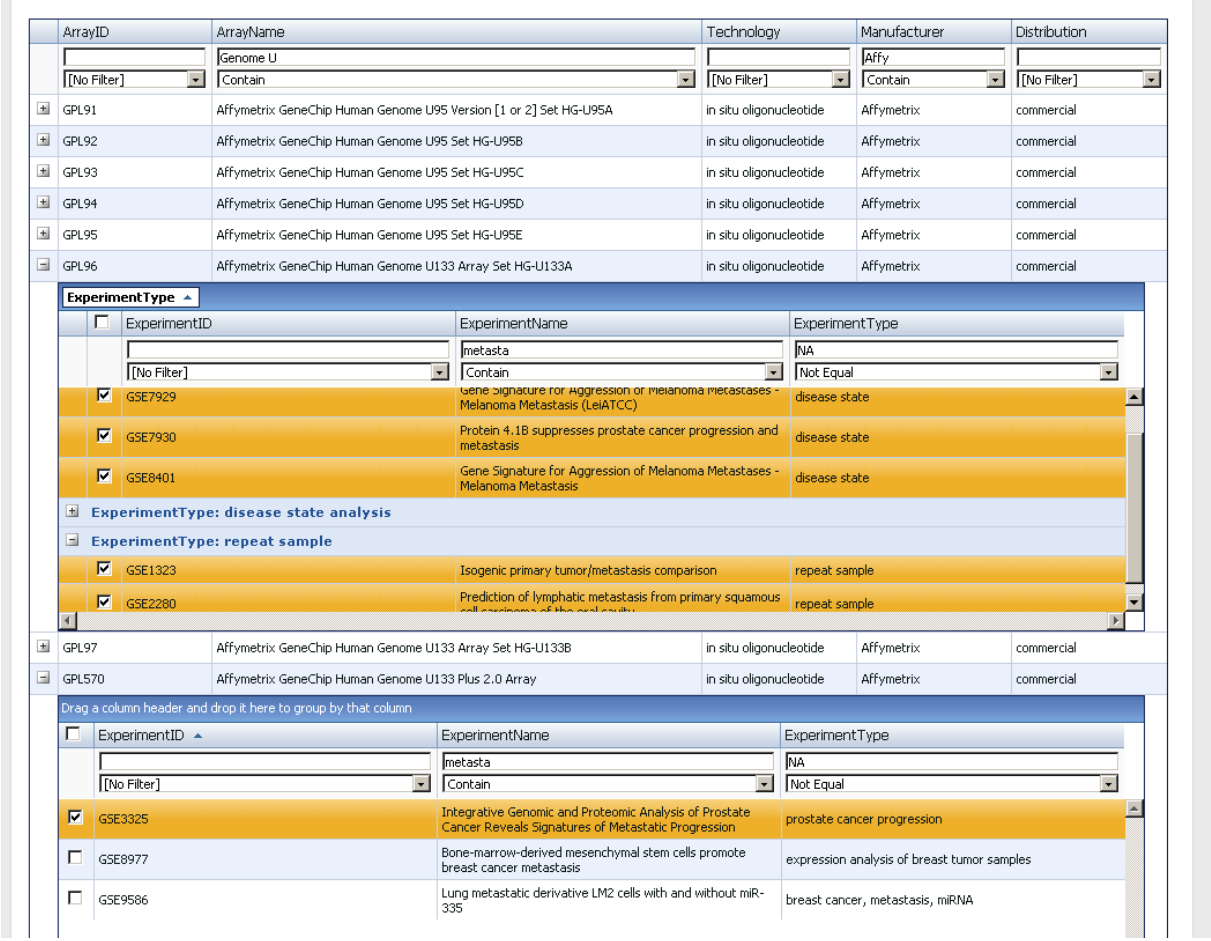


Figure 27 - Search results by "metastasis" keyword.

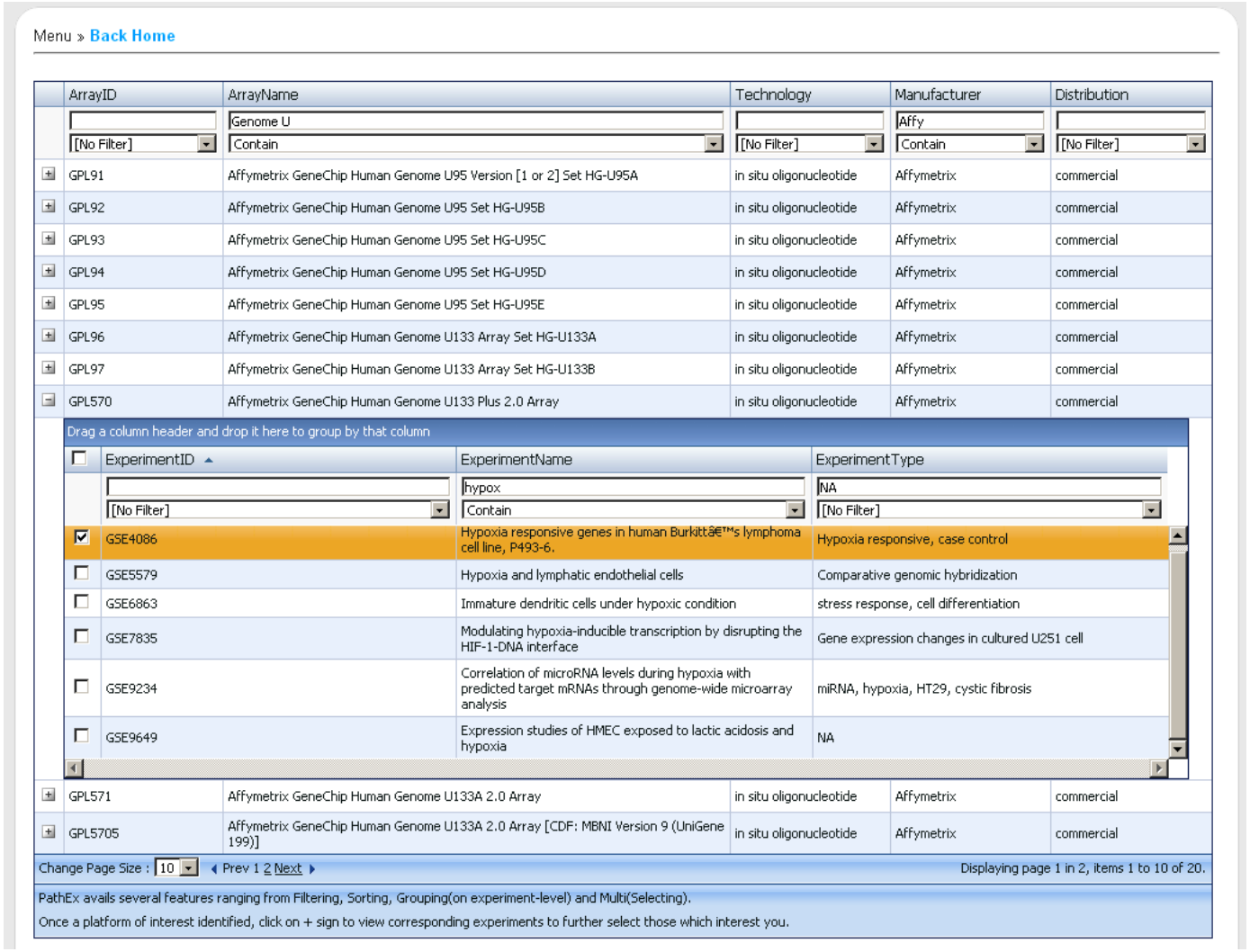


Figure 28 - Search results by "hypoxia" keyword.

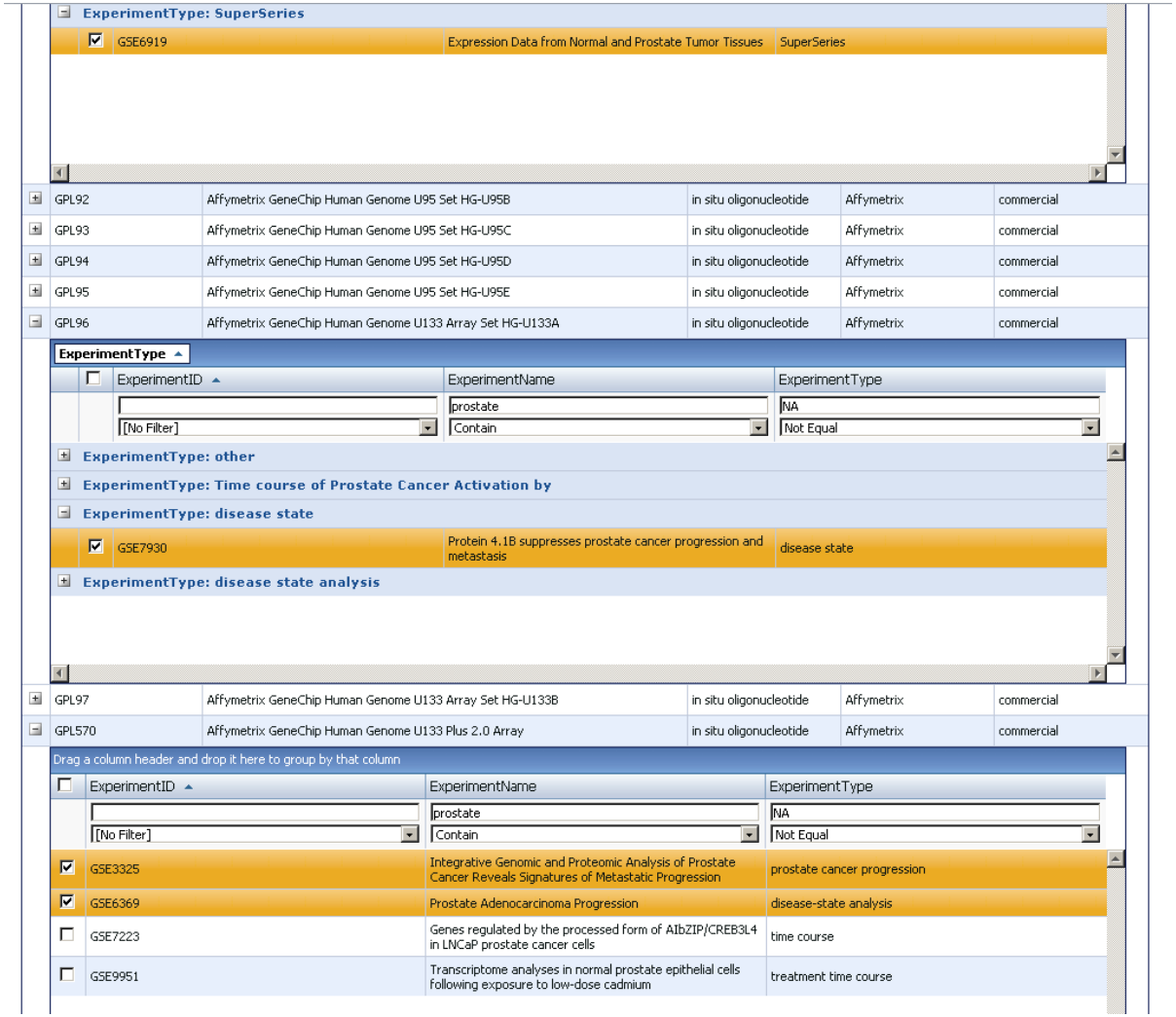


Figure 29 - Search results by "prostate" keyword.

In the second phase, we again used PathEx to generate 14 customized meta-datasets from the 17 original datasets (Table 15).

As we were able to find instantly all datasets initially retrieved manually during three (3) months, we were quite sure that the analysis would highlight as it did previously 183 genes of interest (Figure 30). Out of these genes (in red on the figure), 99 were already known in the literature to be involved in cancer, among which 39 in metastasis, while 21 are related to the response to hypoxia. The other genes of interest found by our methodology are now under investigation to determine their role in hypoxia-induced metastasis.

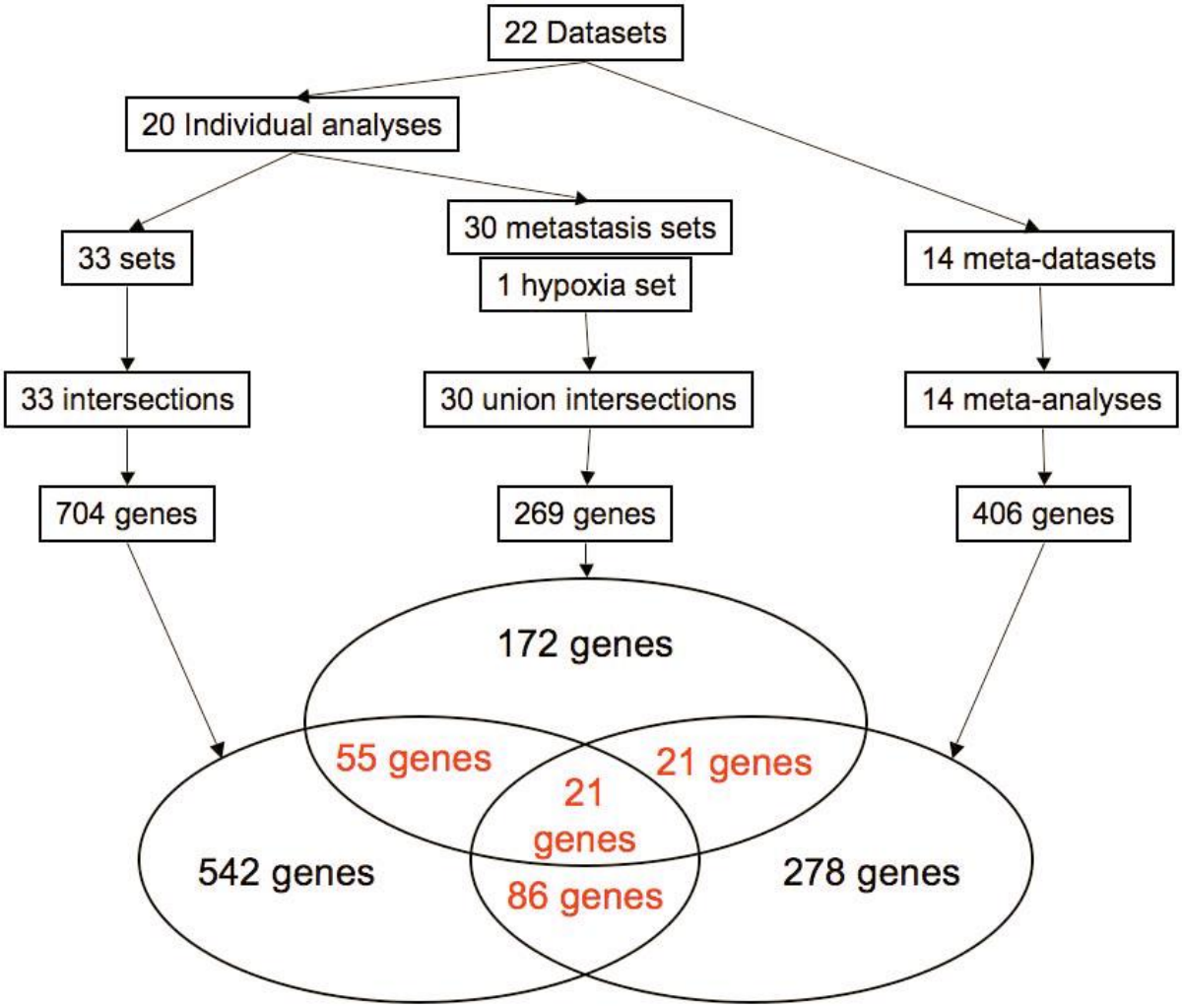


Figure 30 - Venn's diagram of interesting DE genes as revealed by the case study.

The 22 datasets (or sub-datasets) were used to build several combinations in order to run intersections, union intersections and meta-analyses. These three approaches provided 704, 269 and 406 genes respectively.

To validate PathEx, we thought it may be useful to test additionally the PathEx retrieval accuracy and ability. On this step, we considered two published studies, which we knew exactly the datasets initially used to conduct analyses.

In the first study aimed to highlights genes involved in metastasis from Several microarray datasets (Pierre, et al., 2011), the authors presented a new meta-analysis based methodology to pick out genes involved in one or two biological processes from several microarray datasets using a statistic that avoids the definition of an arbitrary threshold, providing statistically-significant results.

As we knew the list of the datasets the used, we searched PathEx using "metastasis" keyword and found exactly the same datasets as the one used in the study (Figure 31).

The specificity of the above two proofing stages is that they were both somehow linked to our research and we did collaborate in them at some extent. The only way to test PathEx capability beyond the studies in which we participated (to avoid) bias, was to make sure that our approach work on third party studies.

We tested again PathEx to see if it can retrieve dataset used in a study on "Transcriptional Profiling after Lipid Raft Disruption in Keratinocytes Identifies Critical Mediators of Atopic Dermatitis Pathways" **(connie et al.)**. To do so, we queried PathEx by Pathways Search with keyword "***Dermatitis***" and we found exactly the same dataset as the original one submitted to GEO in April 2010 by the authors (Figure 32).

After this validation, we think PathEx will be useful for researchers using microarrays, as it will help them to retrieve accurately and in less time datasets specific to their research of interest. PathEx will also provide users with opportunities to re-analyze previously published data by using new analysis methods hoping to unravel new biological phenomena which were not envisioned by the data depositors initially.

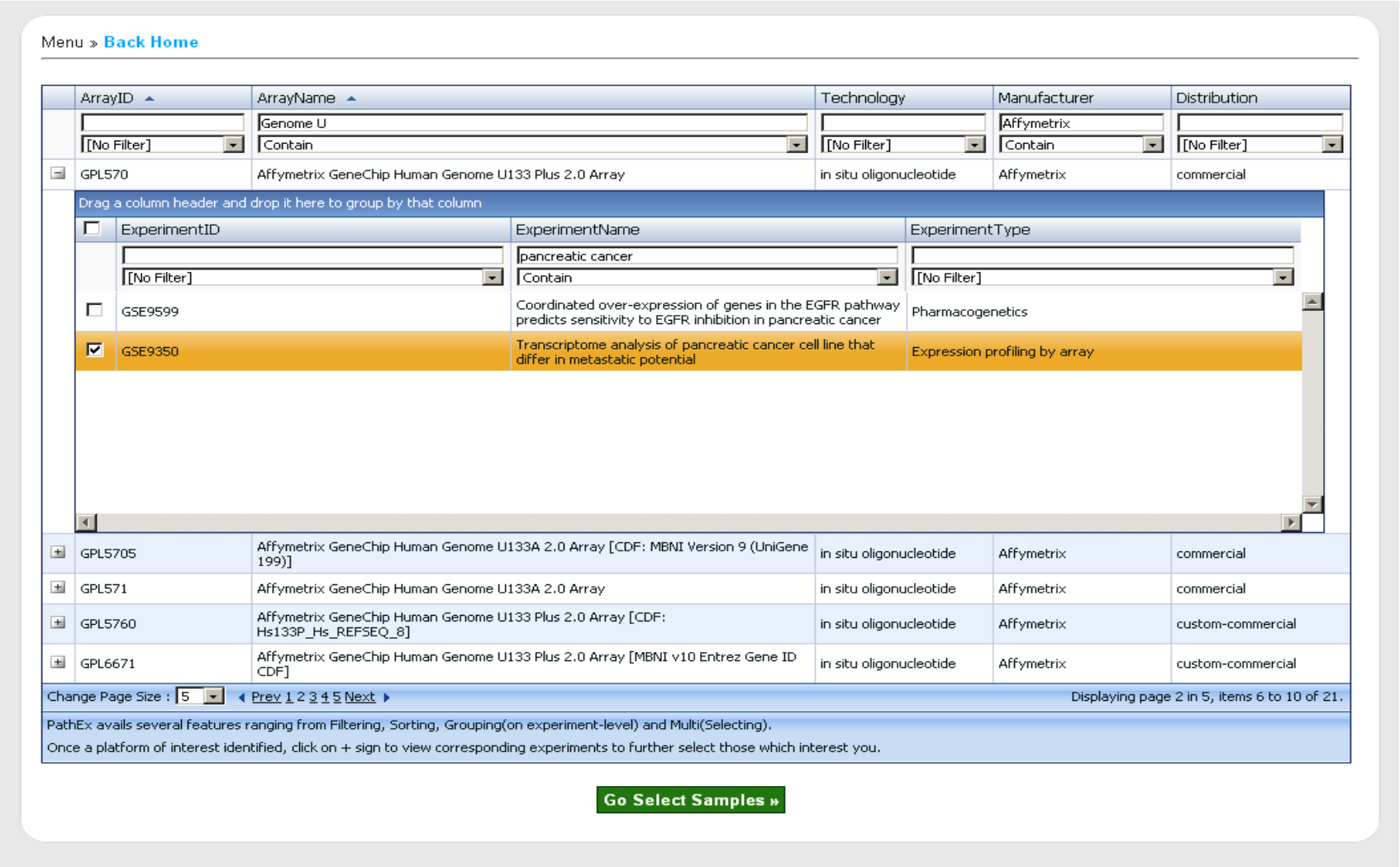


Figure 31 - Pancreatic cancer dataset used to validate the research.

By providing the keyword “metastasis”, PathEx was able to find the exact datasets used in the “Enhanced Meta-analysis study”.

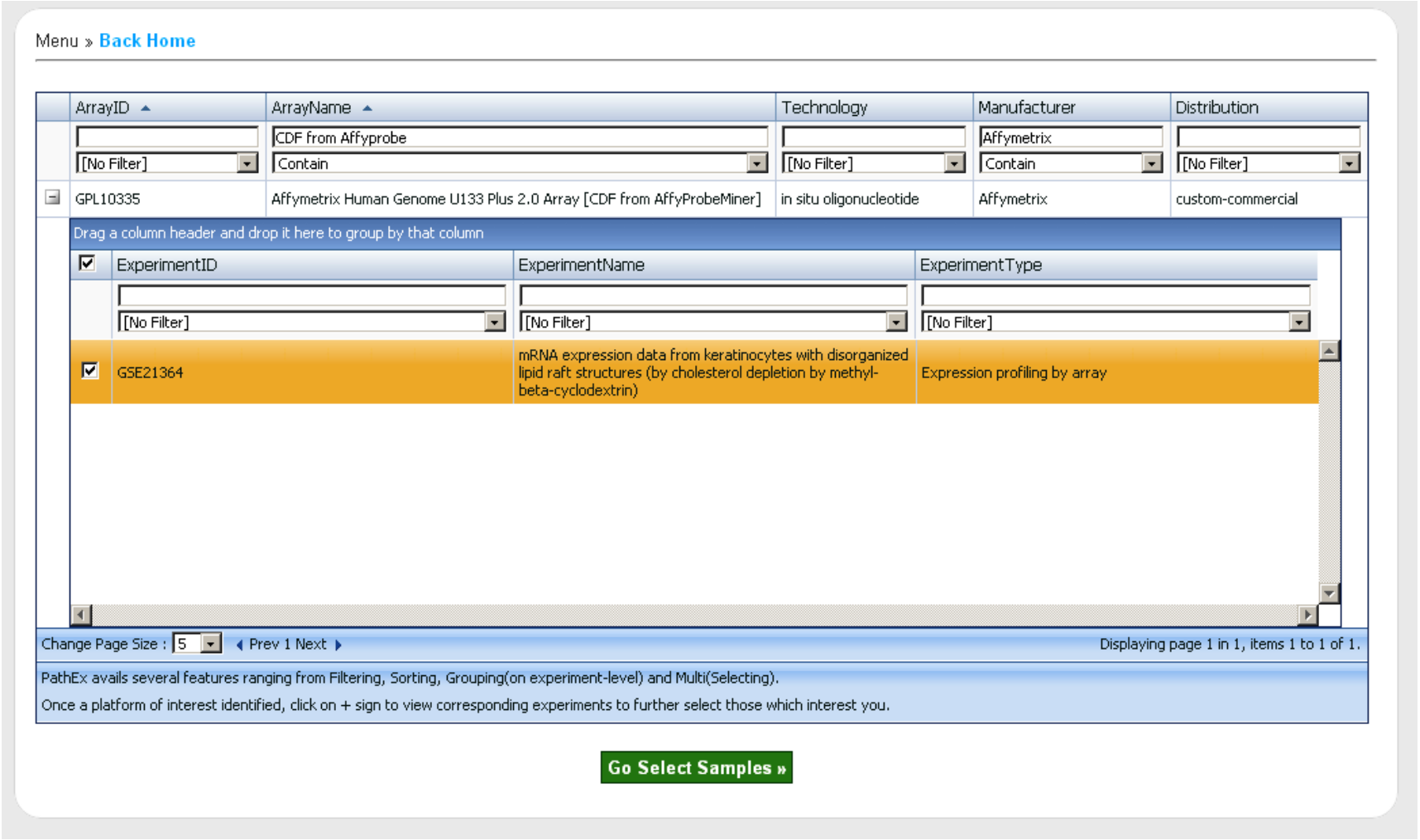


Figure 32 – The dataset used in the Dermatitis study can easily be retrieved by using PathEx.

The single dataset used in the study conducted by Connie in her study was drawn by PathEx by just providing keyword “dermatitisis”.

Additionally, a recent study re-analyzed Atopic Dermatitis previously deposited data (Anne Laure & al., submitted). The study aims to identify new candidate genes involved in the AD pathogenesis and that are also deregulated in MβCD-treated kerationcytes. Several studies have previously demonstrated that Filaggrin, associated with the epidermal barrier structure, is the most important AD susceptibility gene identified to date. Some other candidate genes have been identified to be linked to either the epidermal barrier function or the immune system.

Resulting paper

In this thesis, we reviewed a series of existing microarray analytic methods and approach for filtering technical noise hampering microarray data statistical analysis.

We also presented a novel approach for filtering technical noise, which by combing microarray data drawn from major repositories with biological annotation information from different sources proved to be effective compared to conventional statistical methods.

The proposed solution improves microarray data statistical analysis by providing with users special features to select focused (study-oriented) datasets for further analysis.

The outcome of this approach has been submitted and published as a database issue article in BMC Bioinformatics as shown in the following article manuscript.

The originality of this tool (no such tool exists) made it accepted unanimously by all reviewers without revision requests (except 1 citation we were requested to add).

The first reviewer finds combining the data from both repositories is a nice convenience feature. He has also found the query interface slick and fairly easy to use and he thinks that the article findings are important to those with closely related research interests.

The second reviewer have noticed that PathEx fills a nice gap to get more information between published or by public database container like NCBI GEO or Array Express compared to biochemical analysis tools like KEGG and DAVID to find more information about increased or decreased comparison genes acting in different pathways.

PathEx: A novel multi factors based datasets selector web tool

(as published in BMC Bioinformatics 2010, 11:528doi:10.1186/1471-2105-11-528)

Discussions

In this research study, we present PathEx, a database-principled method for integrating pre-existing biological knowledge with microarray datasets from two major public microarray repositories into a single relational database. From prior knowledge, PathEx extracts “dataset’s metadata keywords” and linked biological annotations, which collectively constitute the datasets in consideration new descriptions (metadata). These dependencies are combined with the datasets to form the PathEx database, which enables expression information to be grouped in novel biologically related datasets capable of predicting and/or validating new biological phenomenon knowledge.

To validate PathEx, we first tested that using random keywords, it was able to find and group all "provided keywords" related datasets exactly in less than 3 minutes. This result, would have taken not less than 3 months when done manually by simply retrieving datasets directly from GEO and ArrayExpress and reading different related literature. However, it was noticed that the use of PathEx should be done with caution as all datasets from GEO and ArrayExpress are heterogeneous in term of research laboratories where they have been conducted, experimental conditions and other factors linked to the datasets producers.

We demonstrated PathEx performance, by using a real, in selecting right datasets on a research study we had initially published (Pierre, et al.) and where we spent almost six months building the datasets of interest. We noticed that our approach performed well while retrieving and instantly building the used datasets and metadata sets.

Then, we further proved that PathEx outperforms major public microarray repositories as it helped us selecting (accurately) and building cancer related metadata sets used to carry out an enhanced meta-analysis (“*Enhanced Meta-analysis highlights genes involved in metastasis from several microarray datasets*”, **Journal of Proteomics Bioinformatics**, volume 4, issue 2, pp. 36-43) in less than 3 minutes, for a task which took manually three (3) months to complete.

We, finally, evaluated PathEx ability to retrieve accurate datasets on "Dermatitis" study (“*Transcriptional profiling after lipid raft disruption in keratinocytes identifies critical mediators of atopic dermatitis pathways*”, **Journal of Investigative Dermatology**, volume 131, pp. 46-58), which was not at all related to our research laboratory.

A closer look at the functionalities PathEx system shows no similarity to other existing tools and repositories, including GEO and ArrayExpress. The difference between all these tools lies in their features for further encoding previously published microarray datasets, retrieving them instantly and generating novel biologically relevant datasets. Practically, the running time of PathEx is approximately less than twenty (20) minutes for the big metadata sets which may exist.

Use of PathEx, however, does require some explicit assumptions about microarray knowledge and some cautions. First, we assume that existing microarray datasets contained in GEO and ArrayExpress were fully and correctly described (annotated). Second, we assume the users have a minimum knowledge on microarray data and methods. Third and finally, issues related to some methods such meta-analysis must be taken into consideration before selecting and building metadata sets for any meta-analysis.

The provided PathEx web application is accessible at http://urbm-cluster.urbm.fundp.ac.be/webapps/pathex. This application includes all necessary functions to test the approach along with several enriched datasets of the Affymetrix platforms type. Given its abilities to improve the rapid and performing retrieval of biologically related datasets, we expect that PathEx will be useful to microarray researchers studying and interpreting (predicting and validating) a wide range of biological phenomena.

Chapter V

Future directions

As discussed in the introduction part of this thesis, microarray data noise occurs at different levels and different proportions. We know that several analytical methods have been proposed to reduce noise from microarray data. However as we demonstrated all along previous chapters, there are other approaches which can be applied to reduce noise using new data annotation systems based on enhanced biological knowledge. This approach being the rationale behind the papers presented in this thesis, proved to be enough effective to improve data retrieval by focusing on particular factor (experiment type, literature, disease, pathway…). The question at this level is "Do PathEx and gViz provide the ultimate solution to all noise issues in microarrays?" The response to this question is quite obvious: "No". Considered alone, PathEx and gViz can somehow be used to reduce noise but, other analytical methods will still be required to significantly reduce microarray data noise.

It is in this vision; we have started integrating different level bioinformatics' solutions to come up with one integrated analysis solution (Figure 36). This new global analysis tool will encompass PathEx as an intelligent dataset generator, DAVID (Dennis, et al., 2003) and PHOENIX (Fabrice Berger, et al., 2009) (for differential analysis) as analysis tool solution (by replacing its data management component by PathEx, the reason here is being the limitation of this component in term of selection features, flexibility and other additional computational advantages.) and MINET package (for co-expression analysis).

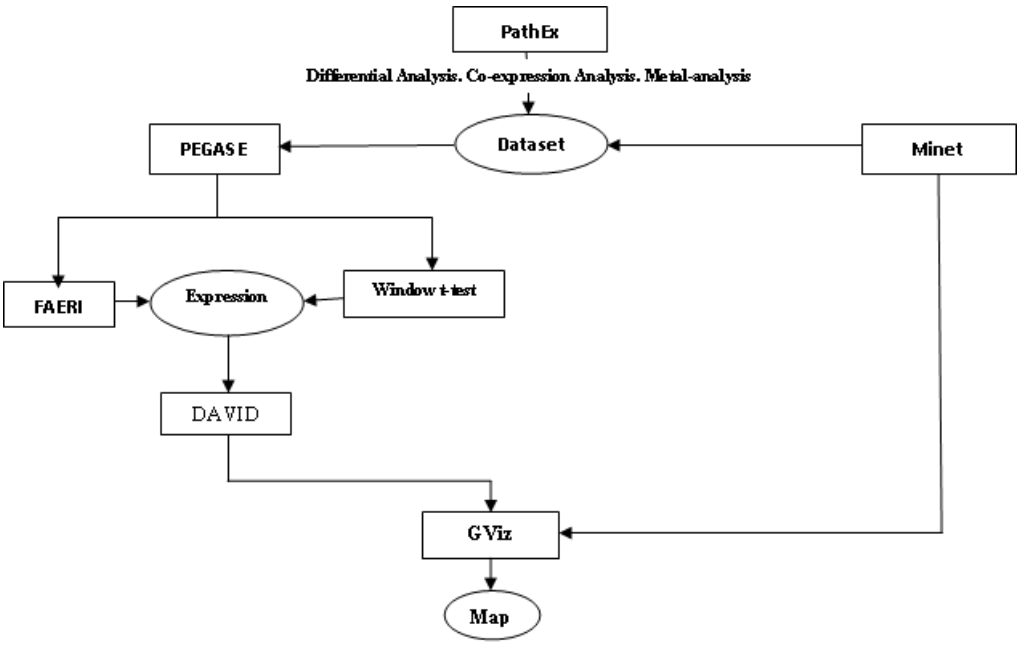


Figure 33 – Overall project future potential directions.

The overall idea of our perspectives is to have a full web-based integrated analysis solution, for users with limited or advanced microarray knowledge. With this solution users will not have to spend precious time looking for and downloading datasets to analyze, reading different analytical methods literature in order to be able to implement them, interpreting the analysis results (Figure 34). A summary sheet, with corresponding nodes (gene-products) outlined by comparing the inferred co-expression networks and maps depicted from KEGG or DAVID on tested genesets (obtained via the use of various differential analysis methods implemented in FAERI or Pegase), will be generated.

Before this project, we initially focused our work on providing appropriate microarray analysis tools with the aim of enhancing prediction of genes of interest. However, these tools do not perfectly outcome full group of genes underlying a full biological phenomenon. We, most of the time, need to complete prediction from these analysis tools by additional information to depict a whole functional and informative gene group. Apart conducting this task experimentally, we can further reduce this work by enhancing our gene group prediction. This can be achieved by availing tool capable of visualizing and manipulating gene networks.

The outcome envisioned in this perspective is to generate potential gene networks to be compared as a validation mechanism. However, the challenges at this level no visualization tool offers mapping features (indicate potential pathways, gene and protein information, molecular function of nodes…).

We proposed in the second part of this thesis a novel gene (-product) co-expression networks visualization tool, which, using PathEx database component touching annotation information such gene, protein and pathway information, allows users to generate, complete and validate gene (-product) co-expression networks of interest.

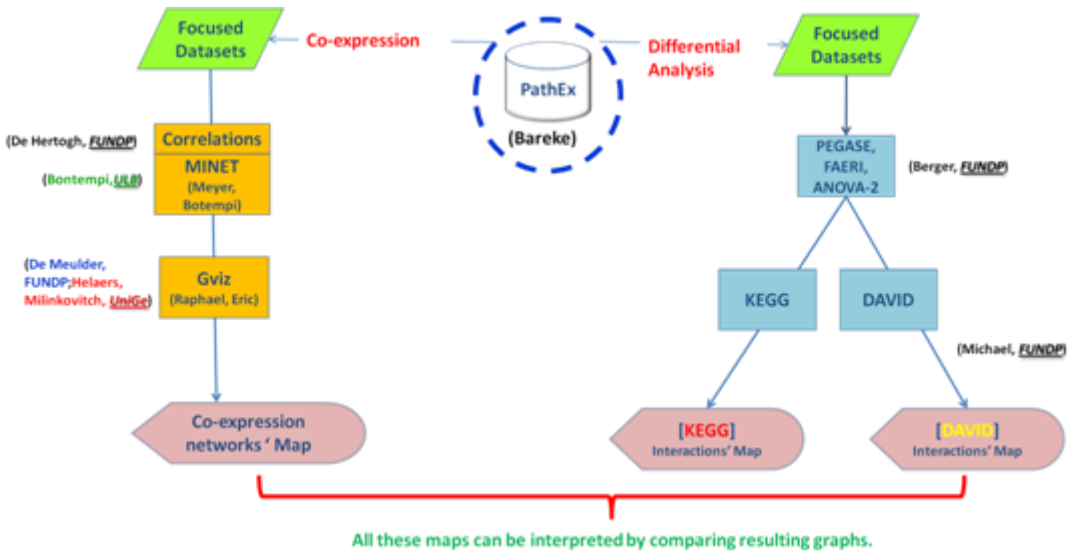


Figure 34 - Overall envisioned analysis pipeline.

gViz: a novel co-expression networks visualization tool

Introduction

Gene expression data have been widely used to study different biological processes and mechanisms. Deriving gene (-product) co-expression networks are often used to extract information about groups of gene (-product)s that are 'functionally' related or co-regulated. However, it has been very difficult to study and fully compare these co-expression networks obtained from gene expression data with known biological networks.

These gene (-product) co-expression networks present however some caveats, which mainly originates from inappropriate technologies used to both visualize and identify the co-expression network components and their interactions involved in a given biological mechanism. Recently, construction of gene (-product) co-expression networks from expression data has recently become an alternative to the standard analytic approaches, such as the detection of differential expression using statistical methods. Therefore, there is a great need of another approach. By combining network theory, biological annotation data, microarray data analysis techniques and advanced graphical features, we developed an innovative, effective and simple to use gene (-product) co-expression networks visualization tool, which can be used to interpret and unravel new biological mechanisms in a simple manner.

Gene (-product) co-expression networks have been used to demonstrate that functionally related gene products are frequently co-expressed across multiple datasets (Fang, et al., 2009). While constructing co-expression networks, it is important to choose the appropriate tool.

In this part, we present a novel gene (-product) co-expression networks visualization tool combining network theory, biological annotation data, microarray data analysis techniques and advanced graphical features.

gViz methods and implementation

Rationale

Merging of network theories, public biological data knowledge and microarray data analysis techniques presents an unexploited opportunity to explore and understand the functionality of genes. It has been assumed that similar patterns in gene expression profiles suggest relationships between genes (Yu, et al., 2003) (Chiang, et al., 2001) and it is important to discover these relationships between co-expressed genes using co-expression matrices from microarray data. While co-expression networks construction may be straightforward, we do limit our work by presenting a visualization tool to investigate the existence of co-expression between genes leaving to other studies the important question of stating whether it is biologically meaningful to represent a gene by a network node and a functional relationship by an edge.

Built around clustering coefficient-based algorithms, gViz relies on a novel approach to visualize and manipulate networks of co-expression interactions among a selection of probesets (each probeset representing a single gene or transcript), based on a set of gene co-expression data stored as an adjacency matrix. This adjacency matrix, representing numerically the relationships between probesets, can be inferred by using co-expression data computational tools such as MINET (Meyer, et al., 2008), an R package allowing computing co-expression relations between probesets and through several microarrays.

The inference of an interaction network is a relatively new domain in bioinformatics. However, several algorithms are available, the following list not being exhaustive: Bayesian approach-based algorithms, such as the package GeneTS (Schafer and Strimmer, 2005), neuronal computation model algorithms, such as the “neuro-fuzzy” (Birney, et al., 2006) approach or mutual information computation based algorithms, such as MINET or qpGraph (Castelo and Roverato, 2009). We chose MINET for several reasons: at each step of the computation process, there is the possibility to choose between alternative methods and to set different parameters; the methods contained within the algorithm are at the edge of current knowledge in the domain; and finally we had the possibility to interact with the developers of the package to refine some steps of the computation to improve what represented a drawback for us to generate adjacent matrix. MINET accepts pretreated microarray data (we recommend to use the GCRMA (Wu, et al., 2004) pretreatment method (for the reasons out of scope of this thesis but proved to be adequate) and provides a co-expression matrix in the form of a GraphML file.

Even if input co-expression matrices are based on probeset identifiers, gViz provides users with advanced features to visualize corresponding interaction networks using

other associated identifiers, such as gene identifiers (EntrezGene id's, Ensembl id's, UniGene (Huynh, et al., 2009) id's and gene symbols), protein identifiers (SwissProt accession numbers) and disease identifiers (OMIM (Rashbass, 1995) id's).

State of the art

There are several programs designed to analyze biological networks, the most known being Cytoscape (Shannon, et al., 2003). Although it is a major first class visualization program, we included in gViz some unique and convenient functionality, first of which is the capacity to compute and display sub networks. We also noticed that gViz is much more user-friendly and easy to use than most of the alternative solutions. Although displaying a huge network containing tens of thousands of nodes and a hundred times more edges is technically possible, it would not humanly be efficient to identify specific parts of that network. It is in this context that gViz proposes different features that allow the user to display only parts of the network of interest. Then user can select one or more identifiers in a list deriving from a provided data set and display the sub-network containing only the relationships it wants to focus on. A slider feature "deepness" provided can be used to adjust gradually the neighborhood of the selected identifiers: if n identifiers are selected and a deepness of d is chosen, the sub-network displayed will include the n identifiers and all their neighbors at a distance of maximum d edges. For example, selecting a single node and a deepness of 2 will display the selected node, its immediate neighbors and neighbors of the latter. It is also possible to set deepness to "maximum", hence displaying all neighbors reachable from the selected identifiers, regardless of their distance. gViz allows also to filter the displayed relationships (edges) by excluding those having a weight under a given threshold (determined during the computational part in MINET; the weight represents the certainty of the selected interaction; i.e., if we consider a pair of nodes i and j, the weight of the arc between them is the maximum of the MRMR (maximum relevance / minimum redundancy) score computed in both directions.). The user can at any time reduce (or increase) the number of edges displayed by eliminating those which are most likely to be false positive. Apart from filtering by edge weight, one can filter the node list by degree (i.e. number of neighbors) and/or by clusters (i.e. sub-networks without connection between them). The clusters are recalculated whenever the user changes the threshold of the edge weights, because they do not take into account edges excluded by the filter. Another prominent feature allows the user to find identifiers by providing annotation criteria (such as a description of a gene or its involvement in a biological process), and generate a sub-network using all or part of the search result.

It is worth mentioning that gViz is, to our knowledge one of the few softwares which can display and analyze GraphML-based networks at the genome scale. The yed graph editor is another solution providing GraphML display capabilities; however it is less powerful and proposes fewer features than gViz. The GraphML format has been

introduced around year 2000 as a common network information exchange format. GraphML format becoming much a major player in network theory, gViz is a serious candidate for widespread GraphML analysis.

Coding

gViz is written in Java and uses JUNG 2 (Java Universal Network / Graph Framework, http://jung.sourceforge.net/) (Fisher and O'Madadhain), a Java library that provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a graph or network. Annotation information displayed on the graph originates from a third-party MySQL database component (this database component derive from PathEx, a web-based, human-oriented, expression array datasets builder for researchers who need to generate organized microarray datasets efficiently and according to their specific needs).

Results

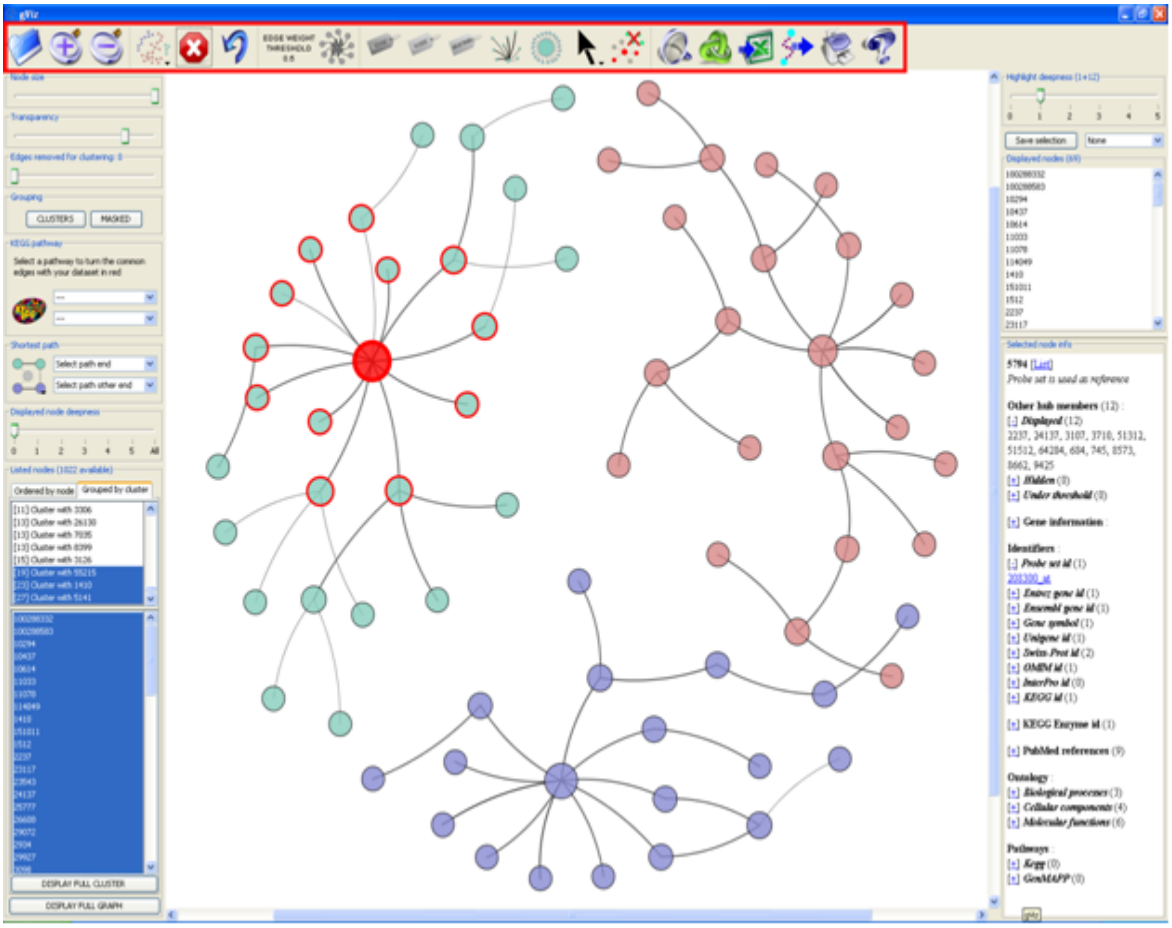


Figure 35 - Example of a co-expression network displayed by gViz.

Networks displayed by gViz (Figure 35) are dynamic and can be manipulated (Figure 36); each node or group of nodes can be moved using the mouse. Several layout algorithms for automatically positioning graph nodes are available and the user is hence provided with the possibility to organize nodes differently depending on the complexity of the actual network. Other 'visual' features include the possibility to change the size and transparency of nodes, to set edge thickness proportional to their weight, or to set node diameter proportional to their degree. Specific nodes can be selected directly on the network graph using the mouse or from a list of displayed identifiers, and their neighbors are highlighted (the number of neighbors is set using a desired distance). User can also generate differently colored clusters by choosing the number of edges to remove to form highly connected sub-networks, where these edges are identified using the Girvan and Newman clustering algorithm (Girvan and Newman, 2002).

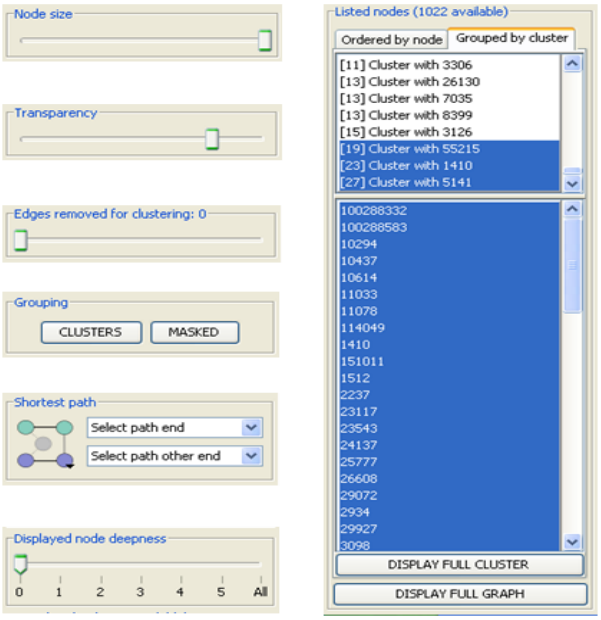


Figure 36 - gViz avails several features to manipulate generated networkS.

When one or more nodes of the network are selected gViz displays a range of annotation information from the displayed sub-network (Figure 35): the list of direct neighbors (displayed or not in the sub-networks, and the neighbors reached by an edge below the threshold set for the weight of edges), and from a biological database: the probeset identifier, information on associated genes, biological processes, cellular components and molecular functions involved (from Gene Ontology (Ashburner, et al., 2000)), proteins (sequences from SwissProt and domains from InterPro (Boeckmann, et al., 2003) (Hunter, et al., 2009)), references in the literature (from PubMed), diseases in which corresponding genes are involved (from

OMIM), chemical reactions and pathways in which these genes are involved (from KEGG and GenMAPP 2 (Salomonis, et al., 2007)).

gViz also has two features that display the shortest path between two nodes, and highlight the edges of the sub-network that can be identified in a pre-selected KEGG pathway.

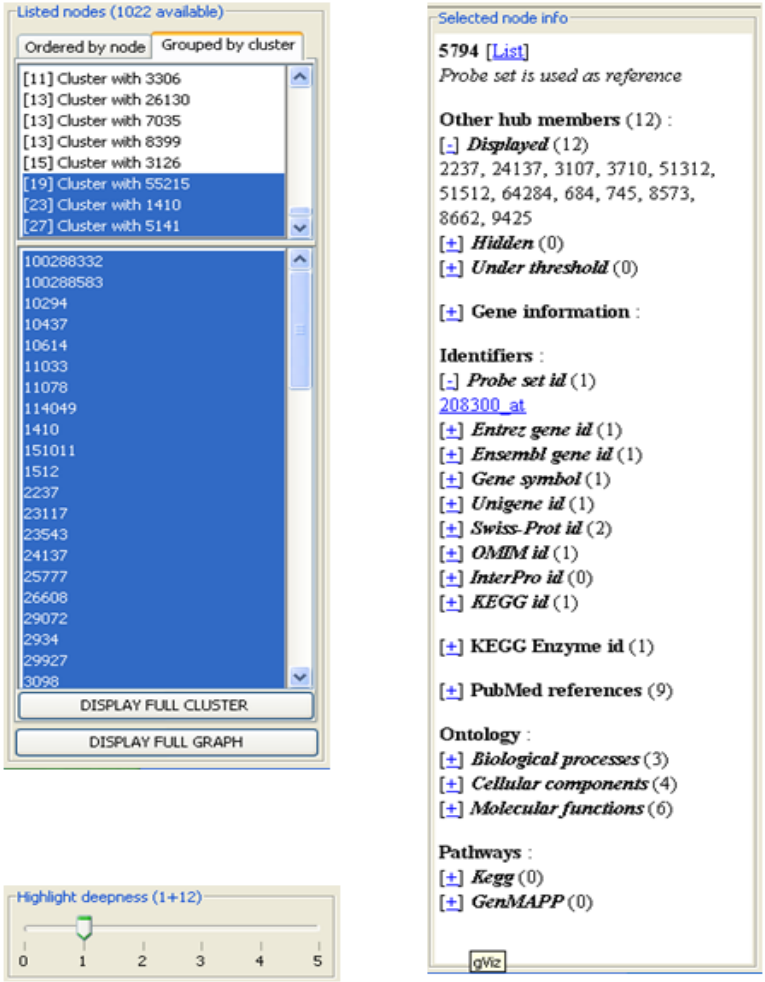


Figure 37 - gViz displays a range of annotation information for generated networks.

Various statistics on the sub-network can be obtained, including histogram (Figure 38) plots providing the user with the degree of network nodes distribution, the diameter of the subnetwork, the weight distribution of the edges, the graph distribution coefficient of clustering (Watts and Strogatz, 1998) and/or cluster size distribution. Finally, the displayed (sub-) network and its statistics can be exported in different data file formats.

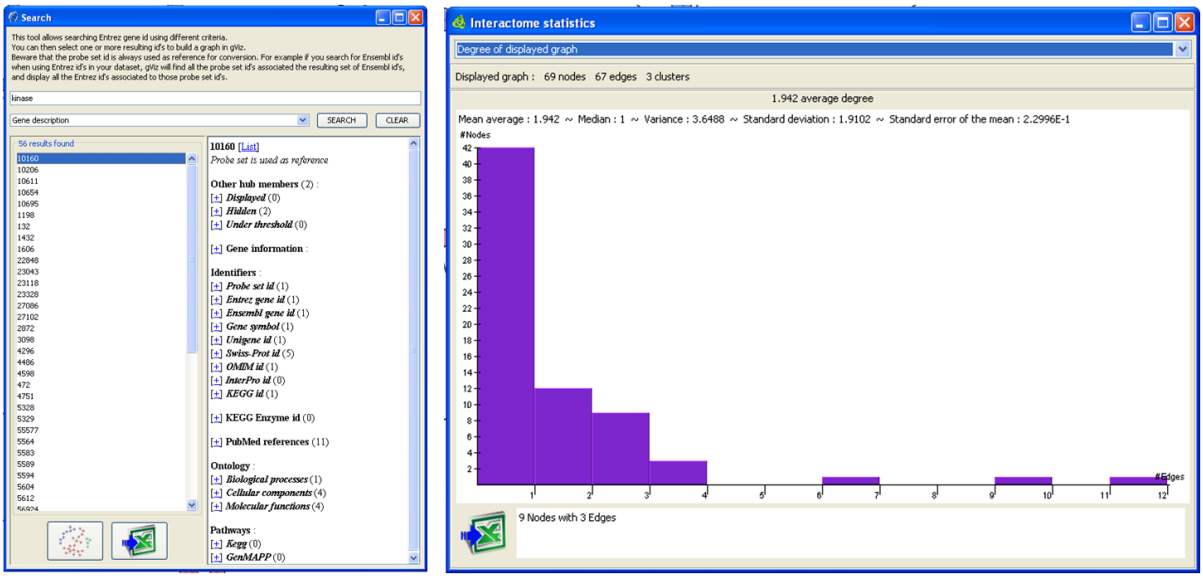


Figure 38 -Example of a kind of a co-expression network' statistics generated by gViz.

gViz methods validation

To validate the effectiveness of gViz tool, we used a dataset initially published in an article we published related to hypoxia and normoxia in cancer cells.

Then the chosen dataset was analyzed and a co-expression matrix computed for visualization.

The obtained matrix was load into gViz and corresponding network generated.

By superimposing the network obtained by gViz in submitting hypoxia key term to the co-expression matrix generated network, we notice that gViz is able to predict a co-expression network without going by co-expression matrices computation.

Concluding remarks

We present in this thesis part, gViz, a novel software for visualization, exploration and analysis of GraphML-based genetic co-expression networks. We described step by step the rationale and utility of such simple but effective tool in case of visualization and inference of gene (-product) networks. gViz provides with convenient built-in filtering and displaying features and connects to an external (PathEx) database component containing information on genes, pathway and other annotation information. Once co-expression matrices obtained, for example using the R package "MINET", gViz may help to explore and/or to map (partially and/or fully) the co-expression relations of gene products in different biological states.

Resulting paper

While developing gViz tool, the aim was to provide with users a tool, which, by using co-expression matrices, predicts gene (-product) co-expression networks.

These co-expression networks being backbones of most interactions underlying (un)known biological phenomena such specific metabolic pathways and/or biological reactions or processes.

The manuscript describing gViz has been submitted in BMC Research Notes as a software issue for review and publication.

We have just received comments from the two appointed reviewers and they sound promising. While the first reviewer needs minor changes on the manuscripts and clarifications before expressing his/her view on the manuscript, the second found the manuscript positive and acceptable for publication after typographic errors correction.

After first revision, the first reviewer accepted gViz for publication and the second accepted it after rephrasing a sentence he found ambiguous. We rephrased the sentence and re-submitted the revised manuscript again for publication.

We hope to have gViz published shortly.

gViz Manuscript

(As July 15, 2011 revision submitted to Bioinformatics Research Notes;

MS: 1142164735546933)

gViz, a novel tool for the visualization of co-expression networks

Raphaël Helaers [1,3,*], Eric Bareke [1, *, §], Bertrand De Meulder [1], Michael Pierre [1], Sophie Depiereux [1], Naji Habra [2], Eric Depiereux [1]

[1]Bioinformatics and Biostatistics unit, Molecular Biology Research Unit (MBRU), Namur Center for Complex Systems (NAXYS), University of Namur (FUNDP)

[2]Research Center in Information Systems Engineering (PReCISE), Faculty of Computing, University of Namur (FUNDP)

[3]Laboratory of Human Molecular Genetics (GEHU), de Duve Institute, Catholic University of Louvain (UCLouvain)

*These authors contributed equally to this work

[§]Corresponding author

Email addresses:

RH: rhelaers@gmail.com

EB: eric.bareke@fundp.ac.be

BDM: bertrand.demeulder@fundp.ac.be

MP: michael.pierre@fundp.ac.be

SD: sophie.depiereux@fundp.ac.be

NH: nha@info.fundp.ac.be

ED: eric.depiereux@fundp.ac.be

Abstract

Background

The quantity of microarray data available on the Internet has grown dramatically over the past years and now represents millions of Euros worth of underused information. One way to use this data is through co-expression analysis. To avoid a certain amount of bias, such data must often be analyzed at the genome scale, for example by network representation. The identification of co-expression networks is an important means to unravel gene to gene interactions and the underlying functional relationship between them. However, it is very difficult to explore and analyze a network of such dimensions. Several programs (Cytoscape, yEd) have already been developed for network analysis; however, to our knowledge, there are no available GraphML compatible programs.

Results

We designed and developed gViz, a GraphML network visualization and exploration tool. gViz is built on clustering coefficient-based algorithms and is a novel tool to visualize and manipulate networks of co-expression interactions among a selection of probesets (each representing a single gene or transcript), based on a set of microarray co-expression data stored as an adjacency matrix.

Conclusions

We present here gViz, a software tool designed to visualize and explore large GraphML networks, combining network theory, biological annotation data, microarray data analysis and advanced graphical features.

Background

The merging of network theories, public biological data knowledge and microarray data analysis techniques presents an unexploited opportunity to explore and understand the functionality of genes. Similar patterns in gene expression profiles have been assumed to suggest relationships between genes (Yu, et al., 2003) and it is important to discover these relationships between co-expressed genes using co-expression matrices from microarray data. While the construction of co-expression networks may be straightforward, we limit our work to the presentation of a visualization tool to search for co-expression between genes and leave to other studies the important question of determining whether it is biologically meaningful to represent a gene by a network node and a functional relationship by an edge.

Several network exploration softwares have been developed these past years; each of them with particular pros and cons. We direct readers to the following review for more details on those programs (Gehlenborg, et al.; Suderman and Hallett, 2007). After testing some of the programs discussed in those reviews, we estimated there was an advantage to build our own visualization program, with several new features included.

Our motivations for developing our program were to allow use of the GraphML (Graph Markup Language) format into a network visualization software and providing a biologists-oriented network exploration solution, both user-friendly and light. To the best of our knowledge, gViz is the only software capable of rendering dynamically GraphML networks. We are aware of the Cytoscape plugin 'Graphmlreader', however it is still in development and seems to be non-functioning at the moment.

The GraphML format is based on XML structure and is therefore ideally suited as a common denominator for all kinds of services generating, archiving, or processing graphs. It supports attributed for nodes and edges, hierarchical graphs and is very flexible [4]. GraphML was developed as modern graph exchange format, suitable in particular for exchange between graph drawing tool and other applications, during the 8th Symposium on graph drawing (GD 2000). It has been developed with the following pragmatic goals in mind: simplicity, generality, extensibility and robustness [5]. We also noticed that this format is very compact and therefore allows for faster transfer or loading into software. One given network uses less storage space when encoded in GraphML format then, for example, in DOT format.

gViz is a novel tool built around clustering coefficient-based algorithms to visualize and manipulate networks of co-expression interactions among a selection of probesets (each probeset representing a single gene or transcript), based on a set of microarray co-expression data stored as an adjacency matrix. This adjacency matrix, representing numerically the relationships between probesets, can be inferred using co-expression data computational tools such as MINET [6], an R package that computes co-expression relations between probesets, and several microarrays.

The inference of interaction networks is a relatively new area in the field of bioinformatics. However, several algorithms are available, including but not limited to the following: Bayesian approach-based algorithms, such as the GeneTS package [7], neuronal computation model algorithms, such as the "neuro-fuzzy" [8] approach, or mutual information computation-based algorithms, such as MINET or qpGraph [9]. We chose MINET for several reasons: at each step of the computation process, it is possible to choose between several methods and to set different parameters; the methods contained in the algorithm are at the cutting edge of current knowledge in the field; and finally we were able to interact with the developers of the package to refine certain steps of the computation. MINET accepts preprocessed microarray data (we chose to use the GCRMA [10] preprocessing method) and provides a co-expression matrix in the form of a GraphML file.

Even if input co-expression matrices are based on probeset identifiers, gViz provides users with advanced features to visualize the corresponding interaction networks using other associated identifiers, such as gene identifiers (Entrez Gene IDs [11], Ensembl IDs [12], UniGene IDs [13] and gene symbols), protein identifiers (SwissProt Accession Codes [14]) and disease identifiers (OMIM IDs [15]).

Implementation

gViz is written in Java and uses JUNG 2 (Java Universal Network / Graph Framework, http://jung.sourceforge.net/), a Java library that provides a common and extensible language for the modeling, analysis and visualization of data that can be represented as a graph or a network. Annotation information displayed on the graph originates from an external microarray database, PathEx [16], a manually-curated web-based expression array dataset builder of human microarray data, for researchers who need to generate organized microarray datasets efficiently and according to their specific needs.

Results and discussion

Comparison with existing software

Several programs have been designed for the analysis of biological networks, the best known of which is Cytoscape [17]. Although it is a major first class visualization program, we chose to develop gViz and to add some unique and convenient functionalities, the first of which is the capacity to compute and display sub-networks using several different built-in filters. This functionality, although potentially biologically important, does not have its counterpart in Cytoscape (see Figure 3). We also noted that gViz is much more user-friendly and easier to use than most of the alternative software packages. Although it is technically possible to display a huge network containing tens of thousands of nodes and a hundred times more edges, it would not be humanly efficient to identify specific parts of that network. In that context, gViz proposes various features that allow the user to display only parts of the network of interest. The user can then select one or more identifiers from a list deriving from a data set provided and display the sub-network containing only the relationships to be studied. A "deepness" slider feature provided can be used to gradually adjust the neighborhood of the selected identifiers: if n identifiers are selected and a deepness of d is chosen, the sub-network displayed will include the n identifiers and all their neighbors at a distance of a maximum of d edges.

For example, selecting a single node and a deepness of 2 will display the selected node, its immediate neighbors and neighbors of the immediate neighbors. It is also possible to set deepness to the "maximum", hence displaying all neighbors reachable from the selected identifiers, regardless of their distance.

gViz can also filter the displayed relationships (edges) by excluding those with a weight under a given threshold (determined during the computational part in MINET; the weight represents the certainty of the selected interaction; i.e., if we consider a pair of nodes i and j, the weight of the arc between them is the maximum of the MRMR (maximum relevance / minimum redundancy) score computed in both directions.). The user can at any time reduce (or increase) the number of edges displayed by eliminating those which are most likely to be false positives.

Apart from filtering by edge weight, one can filter the node list by degree (i.e. number of neighbors) and/or by clusters (i.e. sub-networks without a connection between them). The clusters are recalculated whenever the user changes the threshold of the edge weights, because they do not take into account edges excluded by the filter. Another prominent feature allows the user to find identifiers by providing annotation criteria (such as a description of a gene or its involvement in a biological process), and generate a sub-network using all or part of the search results.

It is worth mentioning that, to our knowledge, gViz is one of the few software packages capable of displaying and analyzing GraphML-based networks at the genome scale. The yEd graph editor [18] is another piece of software able to display GraphML; however, it is less powerful and proposes fewer features than gViz. The GraphML format was introduced around the year 2000 as a common network information exchange format. As such, gViz is a serious candidate for widespread GraphML analysis.

gViz functionalities

Networks displayed by gViz are dynamic and can be manipulated; each node or group of nodes can be moved using the mouse. Several layout algorithms are available to automatically position graph nodes and the user can hence organize nodes differently depending on the complexity of the actual network. Other ‘visual’ features include the possibility to change the size and transparency of nodes, to set edge thickness as a function of their weight, or to set node diameter as a function of their degree. Specific nodes can be selected directly on the network graph using the mouse or from a list of displayed identifiers, and their neighbors are highlighted (the number of neighbors is set using a desired distance). Users can also generate colored clusters differently by choosing the number of edges to remove to form highly connected sub-networks, where these edges are identified using the Girvan and Newman clustering algorithm [19].

When one or more nodes of the network are selected, gViz displays a range of annotation information from the displayed sub-network, such as the list of direct neighbors (displayed or not in the sub-networks) and the neighbors reached by an edge below the threshold set for the edge weight, and from a biological database, such as the probeset identifier, information on associated genes, biological processes, cellular components and molecular functions involved (from Gene Ontology [20]), proteins (sequences from SwissProt and domains from InterPro [21]), references in the literature (from PubMed [22]), diseases in which corresponding genes are involved (from OMIM), chemical reactions and pathways in which these genes are involved (from KEGG [23] and GenMAPP [24]). gViz also has two features that display the shortest path between two nodes and highlight the edges of the sub-network that can be identified in a pre-selected KEGG pathway. Various statistics on the sub-network can be obtained, including histogram plots providing the user with the degree of network node distribution, the diameter of the sub-network, the weight distribution of the edges, the graph distribution coefficient of clustering [25] and/or the cluster size distribution. Finally, the displayed (sub-)network and its statistics can be exported into different image and data file formats.

Case study

In this section, we will give an example of how to collect data, generate a GraphML using Minet and R and explore the resulting graph in gViz.

Data collection

There are several web repositories for microarray data (GEO [26], Array Express [27]). We recommend the use of our database PathEx, for easier and faster data collection.

Network Computation

We use the following packages to compute our networks: GCRMA, Minet and Infotheo (all of which freely available for download in Bioconductor).

Here follows the R code to compute the network from microarray **.cel** files (for R>= R2.10)

library(gcrma)

library(minet)

cel<-list.celfiles()

a<-ReadAffy(filenames=cel)

b<-gcrma(a)

c<-exprs(b)

d<-t(c)

disc<-discretize(d, disc= "equalfreq ", nbins=sqrt(nrow(d)))

mim<-mutinformation(disc)

net<-mrnet(mim)

write.table(net, file= "net.txt ", sep= " \t")

Once these steps are done (which can be long, depending on the computer's speed and dataset size), one can import the network computed into gViz.

gViz exploration

Once the network is loaded in gViz (using the 'open' button {1}, see figure 4 and the gViz manual), the list of the available nodes (genes) is shown in the lower left panel {2}. To display the entire graph, first select the 'circle' layout {3} then click on 'display full graph'{4}. The circle layout is recommended for its low computational needs; rendering a very large network can be extremely resource consuming. However, this step allows for a first overview of the network. At this step, one might want to filter his graph, using the filter on the Minet score {5}, or the filter on the number of neighbors {6}. One can also use the clustering option {7} to group similar nodes, by removing progressively the weakest edges in the graph (controlled by the 'edge removed for clustering' slider {8}). Then, using the selection tool {9}, one highlights the nodes of interest, then hits the 'remove non-selected nodes' button {10}. The resulting sub-graph can then be shown using another, more explicit, layout (for example, 'force-directed').

The different layouts available in gViz have different uses: the 'circle' layout suits best the very big graphs, as it requires less computational power to be displayed. When working on a mid-sized sub-graph (less than 1,000 nodes), the 'Kamada-Kawai' or 'Fruchterman Reingold' layouts are suggested. 'Force directed' or 'Meyer's self organizing' layouts are suited best for small (less than 100 nodes) networks, as they need more computational power to work but better discriminate the edges.

To compute the shortest path between two nodes, first select the 'shortest path' tool {11}, click on the first node of interest then maintain the SHIFT key and click on the second node. gViz automatically computes the shortest path between those two nodes, with respect to the filters possibly applied. This feature can also be controlled via the left panel {11'}. The shortest path can be exported using the 'Export shortest path' button {12}.

It is possible to export the current graph in image (click on 'export network (png)' button {12}) or in various text formats (using the 'export network (text)' button {13}).

To obtain information on a certain node, one simply clicks on its name in the lower right panel {14}, displaying all the information contained in the PathEx database for this particular gene (for more information about PathEx, see [16]).

Conclusions

This manuscript presents gViz, a new software package for the visualization, exploration and analysis of GraphML-based genetic co-expression networks. It has many convenient built-in filtering and displaying features and is connected to an external database containing gene and annotation information. In conjunction with the MINET R package, we use gViz to explore the co-expression relationships of genes in different cellular states.

Availability and requirements

gViz is available at http://urbm-cluster.urbm.fundp.ac.be/webapps/gviz for 32 and 64 bit Windows, MacOS X and Linux/Unix. It requires Java engine 1.6 or higher to run (http://java.com). To connect with the external database PathEx, the port 3306 of the user's computer must be opened. This port may be blocked by a firewall and thus users must ask their network administrator to unblock it for their machine, if necessary. gViz is available under the Open GPL license.

Authors' contributions

RH performed the major part of the coding, as well as debugging and web-publishing of the software. EB developed the PathEx database and took part in the coding and debugging process. BDM took part in the design, debugging and biological testing. MP and SD participated in the biological testing process. NH was involved in the design and implementation process. ED conceived the study and supervised the development and publication. All authors have read and approved the final manuscript.

Acknowledgements and funding

gViz was developed with the support of the Bioinformatics & Biostatistics Lab that provided the computing facilities. We also thank Professor Michel Milinkovitch for providing authorization to use and modify a graphic engine from the Mantis database.

RH was funded by the University of Namur (FUNDP), EB received funding from two sources, namely the University of Namur (FUNDP) and the Belgian Government through the Belgian Technical Cooperation (BTC-CTB, Belgium), BDM works under a Télévie grant (FRS-FNRS, Belgium), MP is financed by the FRIA (FNRS, Belgium) and SD is a grantee of the FRS-FNRS.

Competing interests:

The authors declare that they have no competing interests.

gViz References

1. Yu H, Luscombe NM, Qian J, Gerstein M: **Genomic analysis of gene expression relationships in transcriptional regulatory networks**. *Trends Genet* 2003, **19**(8):422-427.

2. Gehlenborg N, O'Donoghue SI, Baliga NS, Goesmann A, Hibbs MA, Kitano H, Kohlbacher O, Neuweger H, Schneider R, Tenenbaum D *et al*: **Visualization of omics data for systems biology**. *Nat Methods* 2010, **7**(3 Suppl):S56-68.

3. Suderman M, Hallett M: **Tools for visually exploring biological networks**. *Bioinformatics* 2007, **23**(20):2651-2659.

4. **The GraphML file format** [http://graphml.graphdrawing.org/index.html]

5. Brandes U, Marshall MS, North SC: **Graph data format workshop report**. In: *8th International Symposium on Graph Drawing (GD 2000): 2001 2000*: Lecture Notes in Computer Science; 2000: 410-418.

6. Meyer PE, Lafitte F, Bontempi G: **minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information**. *BMC Bioinformatics* 2008, **9**:461.

7. Schafer J, Strimmer K: **An empirical Bayes approach to inferring large-scale gene association networks**. *Bioinformatics* 2005, **21**(6):754-764.

8. Chen CF, Feng X, Szeto J: **Identification of critical genes in microarray experiments by a Neuro-Fuzzy approach**. *Comput Biol Chem* 2006, **30**(5):372-381.

9. Castelo R, Roverato A: **Reverse engineering molecular regulatory networks from microarray data with qp-graphs**. *J Comput Biol* 2009, **16**(2):213-227.

10. Wu Z, Irizarry RA, Gentleman R, Martinez-Murillo F, Spencer F: **A model-based background adjustement for oligonucleotides expression arrays**. *Journal of the American Statistical Association* 2004, **99**(468):909-917.

11. Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: gene-centered information at NCBI**. *Nucleic Acids Res* 2005, **33**(Database issue):D54-58.

12. Hubbard T, Barker D, Birney E, Cameron G, Chen Y, Clark L, Cox T, Cuff J, Curwen V, Down T *et al*: **The Ensembl genome database project**. *Nucleic Acids Res* 2002, **30**(1):38-41.

13. JU P, L W, GD S: **UniGene: a unified view of the transcriptome**. *The NCBI Handbook Bethesda (MD): National Center for Biotechnology Information* 2003.

14. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I *et al*: **The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003**. *Nucleic Acids Res* 2003, **31**(1):365-370.

15. Rashbass J: **Online Mendelian Inheritance in Man**. *Trends Genet* 1995, **11**(7):291-292.

16. Bareke E, Pierre M, Gaigneaux A, De Meulder B, Depiereux S, Berger F, Habra N, Depiereux E: **PathEx: a novel multi factors based datasets selector web tool**. *BMC Bioinformatics* 2010, **11**:528.

17. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome Res* 2003, **13**(11):2498-2504.

18. **yEd- Graph Editor** [http://www.yworks.com/en/products_yed_about.html]

19. Girvan M, Newman ME: **Community structure in social and biological networks**. *Proc Natl Acad Sci U S A* 2002, **99**(12):7821-7826.

20. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium**. *Nat Genet* 2000, **25**(1):25-29.

21. Apweiler R, Attwood TK, Bairoch A, Bateman A, Birney E, Biswas M, Bucher P, Cerutti L, Corpet F, Croning MD *et al*: **The InterPro database, an integrated documentation resource for protein families, domains and functional sites**. *Nucleic Acids Res* 2001, **29**(1):37-40.

22. Guillaume JC: **[PubMed]**. *Ann Dermatol Venereol* 1998, **125**(6-7):467-468.

23. Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M: **KEGG: Kyoto Encyclopedia of Genes and Genomes**. *Nucleic Acids Res* 1999, **27**(1):29-34.

24. Salomonis N, Hanspers K, Zambon AC, Vranizan K, Lawlor SC, Dahlquist KD, Doniger SW, Stuart J, Conklin BR, Pico AR: **GenMAPP 2: new features and resources for pathway analysis**. *BMC Bioinformatics* 2007, **8**:217.

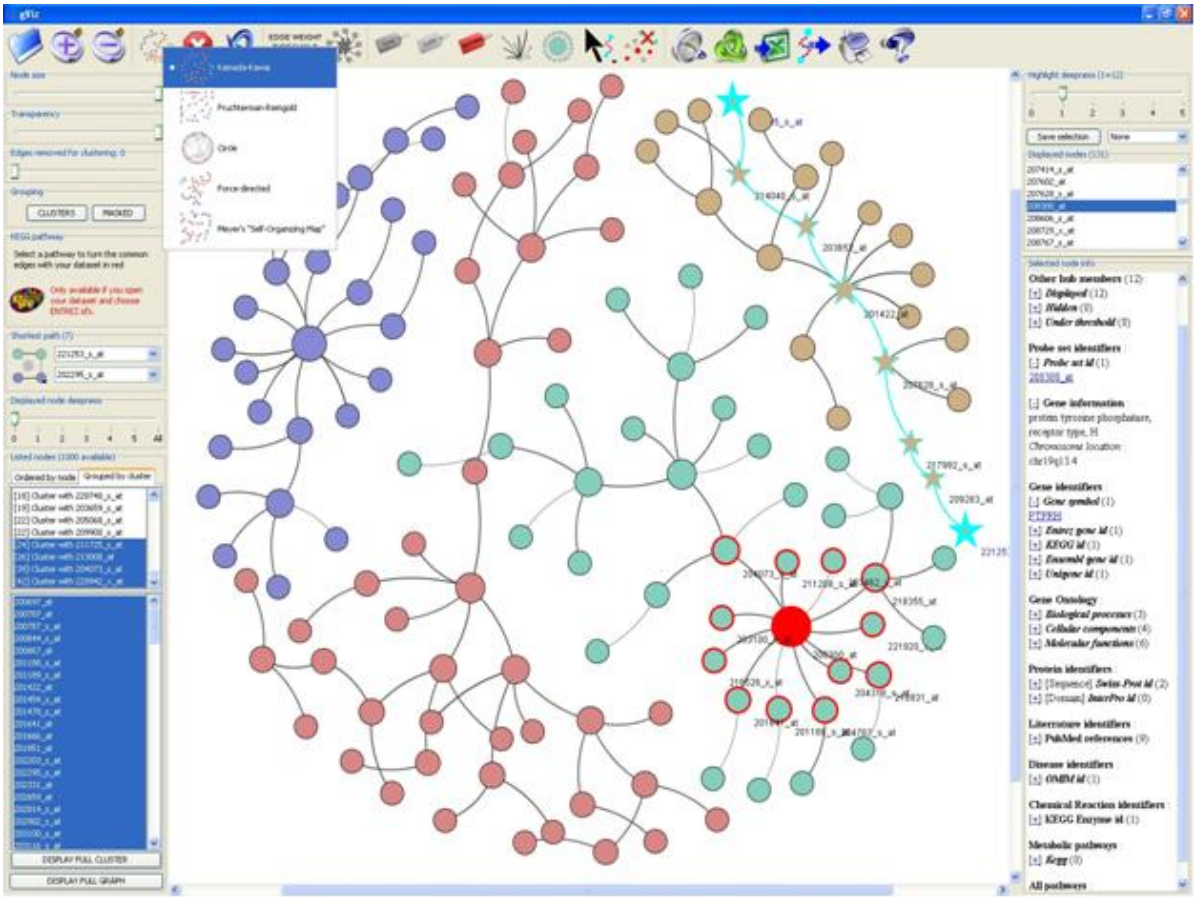
25. Watts DJ, Strogatz SH: **Collective dynamics of 'small-world' networks**. *Nature* 1998, **393**(6684):440-442.

26. Barrett T, Troup DB, Wilhite SE, Ledoux P, Rudnev D, Evangelista C, Kim IF, Soboleva A, Tomashevsky M, Edgar R: **NCBI GEO: mining tens of millions of expression profiles--database and tools update**. *Nucleic Acids Res* 2007, **35**(Database issue):D760-765.

27. Parkinson H, Kapushesky M, Shojatalab M, Abeygunawardena N, Coulson R, Farne A, Holloway E, Kolesnykov N, Lilja P, Lukk M *et al*: **ArrayExpress--a public database of microarray experiments and gene expression profiles**. *Nucleic Acids Res* 2007, **35**(Database issue):D747-750.

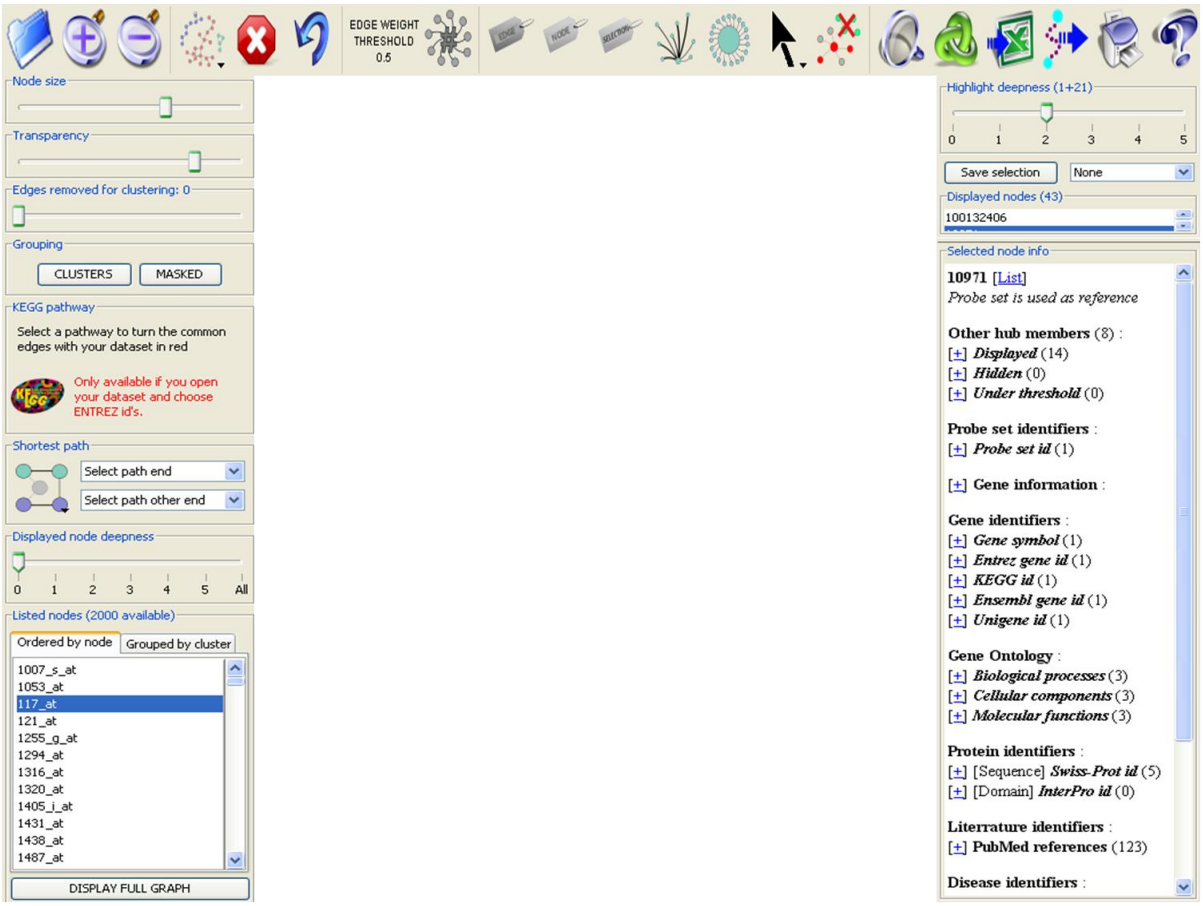
Figures

Figure 1 - Screenshot of the general interface of gViz



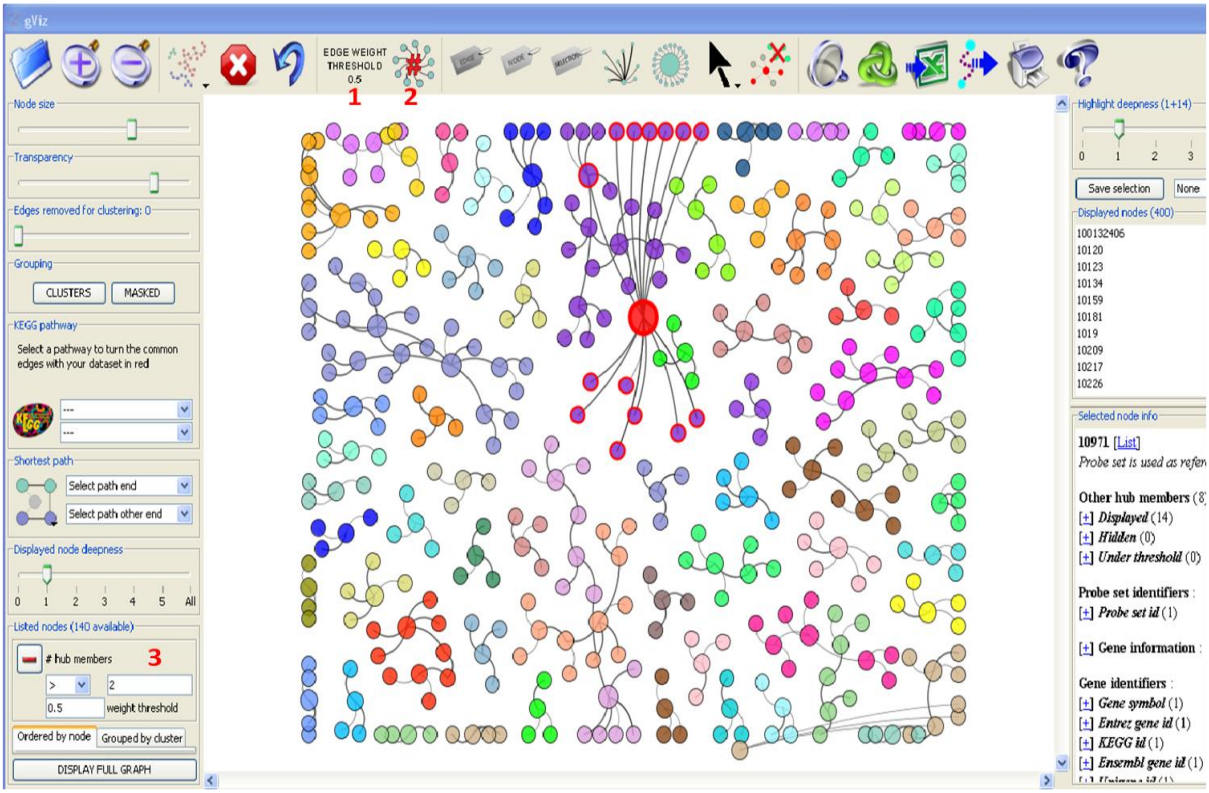
This is the interface of a typical analysis in gViz. The center panel displays the current network. The top panel contains the visual functionalities. The left panel contains the different sliders as well as the clustering and comparison tools. Finally, the right panel shows the information contained in PathEx on the nodes selected in the current network.

Figure 2 - Zoom of the top, left and right panels



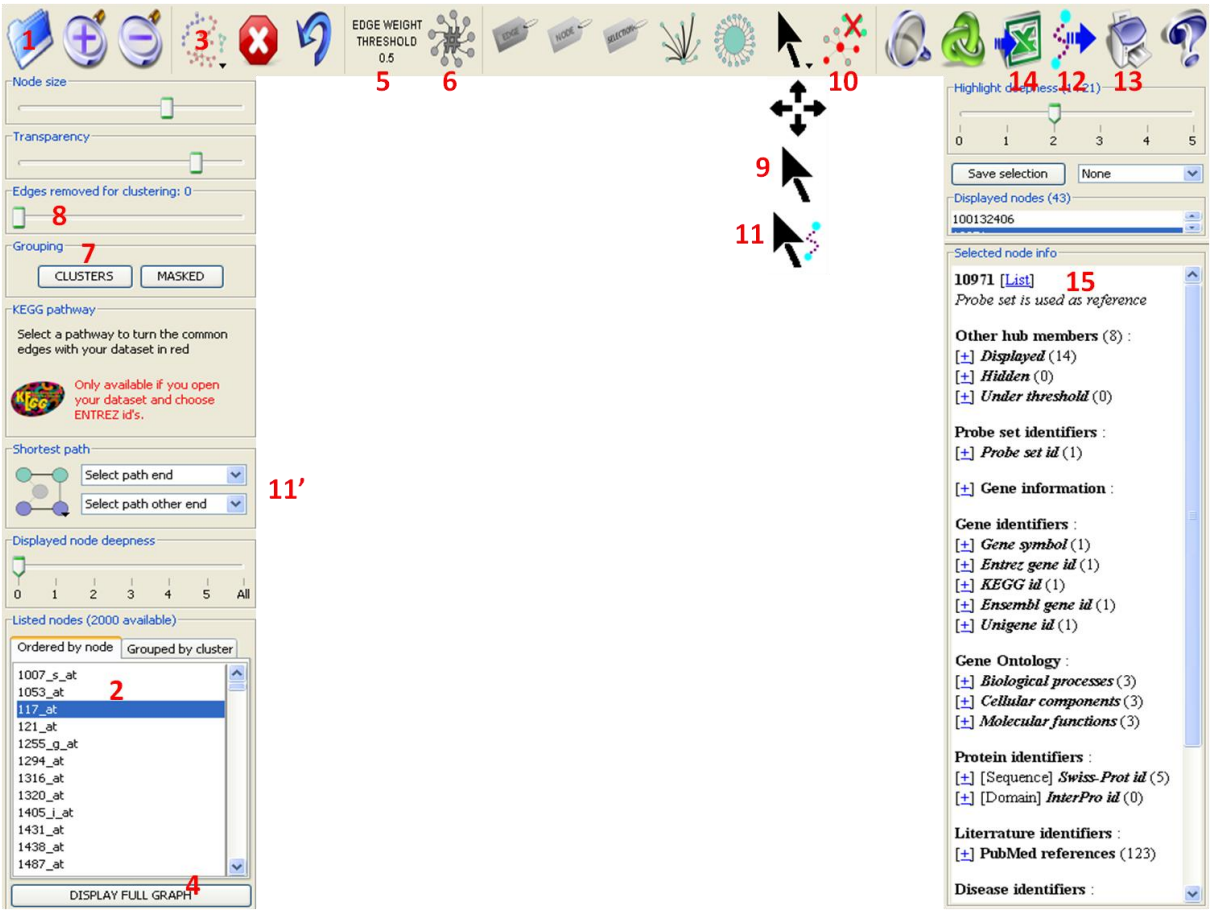
Detailed view of the top panel where are grouped the visual tools of gViz, left panel containing the sliders along with the clustering and comparison tools and right panel presenting the information from PathEx on the selected genes, as well as slider to adjust the selection deepness.

Figure 3 – Case Study of gViz



This figure displays a network filtered by the number of neighbors (> 2) and the Minet score (score > 0.5). The filters are triggered by the buttons in {1} and {2}, and controls for the filters are accessed in the left lower panel {3}.

Figure 4 – Filters in gViz



Detailed view of the gViz interface and locations of the tools mentioned in the 'case study' section. Legend: 1 – Open button; 2- Lower left panel, list of nodes available; 3- Layout control button; 4- Display full graph button; 5- Filter on edges scores; 6- Filter on number of neighbors; 7- Clustering button; 8- Edges removed for clustering control slider; 9- Selection tool; 10- Remove non-selected nodes buttons; 11- Shortest path tool; 12- Export shortest path button; 13- Export network (png); 14- Export network (text); 15- Lower right panel, information on selected nodes.

Chapter VI

Summary & Conclusions

DNA microarray technology is a powerful research tool that enables the global measurement of changes between paired RNA samples. While strong biological inferences can be made from microarray data, they must be made within the proper biological and experimental context. This is because DNA microarrays capture a static view of a dynamic molecular event. This static view challenges researchers to unravel meaningful biological changes from the associated noise (due to sample acquisition, target labeling, microarray processing, etc.)

DNA microarrays are notorious for generating noisy data. A common strategy for reducing or eliminating the effects of noise is to perform many experimental replicates.

This approach is often costly and sometimes impossible given limited resources; thus, other approaches are needed which increase accuracy at no additional cost. One inexpensive source of microarray replicates comes from prior work: to date, data from hundreds of thousands of microarray experiments and biological information are in the public domain.

Although these data assay a wide range of conditions, they cannot be used directly to inform any particular experiment and are thus ignored by most analysis methods.

We envision PathEx as a means to maximize the potential for biological discovery from biological and microarray pre-knowledge, and we provide it as a freely available software package that is immediately applicable to any human microarray study.

Several analytical methods have been proposed to resolve this noise issues. They mainly focus on estimating noise assuming that the level of the latter is signal intensity dependent. To achieve this estimation they propose to use a set of replicate arrays with varying degrees of experimental differences. However, a plethora of different analytical methods (with different outcomes) has led to the question of reproducibility of gene expression data, hampering hopes researchers have had with the advent of microarray technology.

As the main goal of any biological experiment is to understand a specific mechanism, we proposed in this thesis another approach, which coupled with existing microarray analytical methods, may implicitly and substantially filter technical noise (biological noise being unavoidable and even informative).

The advantage of one of the presented approach (PathEx) is that it resolves at the same time the issue of lack of replicates in addition to the provision of possibility to design and generate automatically user and focus driven datasets.

PathEx exploits thousands of public availed gene expression data from major public repositories, re-annotate them using an expert system combining various biological information sources (gene, protein, pathway, disease, ontology, interaction…), structure and organize them into a web based relational database.

By proceeding this way, we created a new microarray knowledge base, which can be used by researchers with limited financial means (considering the cost of microarray technology) and with limited knowledge of microarrays (users do not need to know the type and structure of various sources integrated into the outcome solution of this approach).

We presented PathEx, a database-principled, biological data-driven approach, to increase the power of microarray experiments in interpreting (predicting and validating) differentially expressed genes and biological phenomena related gene groups.

We achieved this by designing and implementing a novel web-based microarray database tool combining together microarray datasets form GEO and ArrayExpress and a series of biological annotation information to further characterize those datasets.

We demonstrated all along this thesis the power of this tool and we believe that use of the presented tool will help to interpret, validate and further develop biological new hypotheses without the need to conduct new experiments.

On the other side, we detailed how by using a newly developed, effective network visualization tool such as gViz (with its advanced mapping capabilities), we can easily interpret biological phenomenon candidates with the support of knowledge contained in various biological sources integrated into PathEx database backend components.

We showed that there are still ways to improve predictive and validation power of existing analysis methods by considering new approaches that have never been envisioned up to now. We believe the tools developed (and provided), without being ultimate solutions, contribute to better advancement of biomedical research.

There are, however, common drawbacks that researchers face when interpreting DNA microarray results. The recognition of these drawbacks will help minimize false leads and maximize the biological meaning of the resulting microarray analysis.

Among these drawbacks we may mention, without being exhaustive, that:

- Although replicate experiments offer statistical benefits, researchers should be aware that different levels of replication provide different answers.

- DNA microarray results generally include genes that are considered "differentially expressed" between RNA samples and genes that are considered unchanged between samples. This distinction is determined by statistical tests using the spot intensity value for each gene. Although the use of statistics is recommended for defining differential expression, the statistical confidence of differential expression calls may not always reflect the biological relevance.

- While many researchers focus on the handling of DNA microarrays, they should also recognize that the design of an experiment is as important as the implementation of the experiment.

- The goal of any measurement tool is to provide an estimate of quantitative truth. In case of microarrays, this truth is of differential gene expression. And, like other quantitative tools, DNA microarray measurements are typically associated with an estimate of measurement error.

- The proper interpretation of DNA microarray results should always be done within the context of biological information, experimental design, and statistical output. If pursued independently, each individual path could result in misleading biological interpretation.

Bibliography

List of thesis references

Affymetrix (2001) Affymetrix Microarray Suite User Guide version 5.0.

Affymetrix, b. (2002) Statistical Algorithms used in the MAS 5.0 software, *Affymetrix*.

Amaratunga, D., Cabrera, J., Fernholz, L., Morgenthaler, S., Ronchetti, E. and Stahel, W. (2001) Outlier resistance, standardization and modeling issues for DNA microarray data, *Statistics in the Sciences*.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. and Sherlock, G. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nat Genet*, **25**, 25-29.

Ashurst, J.L., Chen, C.K., Gilbert, J.G., Jekosch, K., Keenan, S., Meidl, P., Searle, S.M., Stalker, J., Storey, R. and Trevanion, S. (2005) The Vertebrate Genome Annotation (Vega) database, *Nucleic Acids Res*, **33 Database**, D459 - 465.

Barrett, T. and Edgar, R. (2006) Gene expression omnibus: microarray data storage, submission, retrieval, and analysis, *Methods Enzymol*, **411**, 352 - 369.

Barrett, T., Troup, D.B., Wilhite, S.E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I.F., Soboleva, A., Tomashevsky, M., Marshall, K.A., Philippy, K.H., Sherman, P.M., Muertter, R.N. and Edgar, R. (2009) NCBI GEO: archive for high-throughput functional genomic data, *Nucleic Acids Res*, **37**, D885 - 890.

Bell, D.C., Thomas, W.K., Murtagh, K. and Glover, W.R. (2010) DNA Sequencing with TEM, *Microsc. Microanal.*, **16** 1768-1769.

Ben-Gal, I., Shani, A., Gohr, A., Grau, J., Arviv, S., Shmilovici, A., Posch, S. and Grosse, I. (2005) Identification of transcription factor binding sites with variable-order Bayesian networks, *Bioinformatics*, **21**, 2657-2666.

Berger, F., De Hertogh, B., Pierre, M., Gaigneaux, A. and Depiereux, E. (2008) The "Window t test": a simple and powerful approach to detect differentially expressed genes in microarray datasets, *Central European Journal of Biology*, **3**, 327 - 344.

Berger, F., De Meulder, B., Gaigneaux, A., Depiereux, S., Bareke, E., Pierre, M., De Hertogh, B., Delorenzi, M. and Depiereux, E. Functional Analysis: Evaluation of Response Intensities - Tailoring ANOVA for Lists of Expression Subsets, *BMC Bioinformatics*, **11**, 510.

Birney, E., Andrews, D., Bevan, P., Caccamo, M., Cameron, G., Chen, Y., Clarke, L., Coates, G., Cox, T. and Cuff, J. (2004) Ensembl 2004, *Nucleic Acids Res*, **32 Database**, D468 - 470.

Birney, E., Andrews, D., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cox, T., Cunningham, F., Curwen, V. and Cutts, T. (2006) Ensembl 2006, *Nucleic Acids Res*, **34 Database**, D556 - 561.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C. and Phan, I. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, *Nucleic Acids Res*, **31**, 365 - 370.

Bolstad, B.M. (2002) Comparing the effects of background, normalization and summarization on gene expression estimates.

Bolstad, B.M., Irizarry, R.A., Astrand, M. and Speed, T.P. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics*, **19**, 185-193.

Bolstad, B.M., Irizarry, R.A., Astrand, M. and Speed, T.P. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics*, **19**, 185 - 193.

Bozinovic, G., Sit, T., Hinton, D. and Oleksiak, M. Gene expression throughout a vertebrate's embryogenesis, *BMC Genomics*, **12**, 132.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., Gaasterland, T., Glenisson, P., Holstege, F.C., Kim, I.F., Markowitz, V., Matese, J.C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J. and Vingron, M. (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data, *Nat Genet*, **29**, 365-371.

Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P. and Lara, G.G. (2003) ArrayExpress--a public repository for microarray gene expression data at the EBI, *Nucleic Acids Res*, **31**, 68 - 71.

Burbeck, S. (1992) Application Programming in Smalltalk-80: How to use Model-View-Controller (MVC). *University of Illinois in Urbana-Champaign (UIUC)*.

CAMPBELL, J.D., SPIRA, A. and LENBURG, M.E. (2011) Applying gene expression microarrays to pulmonary disease, *Official Journal of Asian Pacific Society of Respirology*, **16**, 407–418.

Castelo, R. and Roverato, A. (2009) Reverse engineering molecular regulatory networks from microarray data with qp-graphs, *J Comput Biol*, **16**, 213-227.

Chen, S., Phillips, M.F., Cerrina, F. and Smith, L.M. (2009) Controlling Oligonucleotide Surface Density in Light-Directed DNA Array Fabrication, *Langmuir*, **25**, 6570-6575.

Chiang, D.Y., Brown, P.O. and Eisen, M.B. (2001) Visualizing associations between genome sequences and gene expression data using genome-mean expression profiles, *Bioinformatics*, **17**, S49-S55.

Cooper, S. and Shedden, K. (2003) Microarray analysis of gene expression during the cell cycle, *Cell & Chromosome*, **2**, 1.

Cui, X., Hwang, J.T., Qiu, J., Blades, N.J. and Churchill, G.A. (2005) Improved statistical tests for differential gene expression by shrinking variance components estimates, *Biostatistics*, **6**, 59 - 75.

De Hertogh, B., De Meulder, B., Berger, F., Pierre, M., Bareke, E., Gaigneaux, A. and Depiereux, E. A benchmark for statistical microarray data analysis that preserves actual biological and technical variance, *BMC Bioinformatics*, **11**, 17.

De Hertogh, B., De Meulder, B., Berger, F., Pierre, M., Bareke, E., Gaigneaux, A. and Depiereux, E. (2009) A benchmark for statistical microarray data analysis that preserves actual biological and technical variance, *BMC Bioinformatics*, **11**, 17.

De Hertogh, B., De Meulder, B., Berger, F., Pierre, M., Bareke, E., Gaigneaux, A. and Depiereux, E. (2010) A benchmark for statistical microarray data analysis that preserves actual biological and technical variance, *BMC Bioinformatics*, **11**, 17.

Dennis, G., Sherman, B.T., Hosack, D.A., Yang, J., Gao, W., Lane, H.C. and Lempicki, R.A. (2003) DAVID: Database for Annotation, Visualization, and Integrated Discovery, *Genome Biol*, **4**, P3.

Drews, J.r. (2000) Drug Discovery: A Historical Perspective, *Science*, **287**, 1960-1964.

Eisen, M.B. and Brown, P.O. (1999) DNA arrays for analysis of gene expression, *Methods Enzymol*, **303**, 179-205.

Fabrice Berger, B.D.H., Pierre, M., Bareke, E., Gaigneaux, A. and Eric, D. (2009) PHOENIX, a web interface for (re)analysis of microarray data, *Central European Journal of Biology*, **4**, 15.

Fang, Y. Resonant Waveguide Grating Biosensor for Microarrays.

Fang, Y.J., Lu, Z.H., Wang, G.Q., Pan, Z.Z., Zhou, Z.W., Yun, J.P., Zhang, M.F. and Wan, D.S. (2009) Elevated expressions of MMP7, TROP2, and survivin are associated with

survival, disease recurrence, and liver metastasis of colon cancer, *Int J Colorectal Dis*, **24**, 875 - 884.

Fisher, D. and O'Madadhain, J. JUNG 2: the Java Universal Networks/Graph API

Flicek, P., Aken, B.L., Beal, K., Ballester, B., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cunningham, F. and Cutts, T. (2008) Ensembl 2008, *Nucleic Acids Res*, **36 Database**, D707 - 714.

Fodor, S.P.A., Rava, R.P., Huang, X.C., Pease, A.C., Holmes, C.P. and Adams, C.L. (1993) Multiplexed biochemical assays with biological chips, *Nature*, **364**, 555-556.

Ganguly, S., Paul, I. and Mukhopadhayay, S.K. (2010) DNA microarray technique for diagnosis of various animal infections – a brief introductory preview, *Indian Pet Journal*, **5**.

Gehlenborg, N., O'Donoghue, S.I., Baliga, N.S., Goesmann, A., Hibbs, M.A., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D. and Gavin, A.C. Visualization of omics data for systems biology, *Nat Methods*, **7**, S56-68.

Girvan, M. and Newman, M.E. (2002) Community structure in social and biological networks, *Proc Natl Acad Sci U S A*, **99**, 7821-7826.

Goddard, M.J. and Hinberg, I. (1990) Receiver operator characteristic (ROC) curves and non-normal data: an empirical study, *Stat Med*, **9**, 325-337.

Goeman, J.J. and Buhlmann, P. (2007) Analyzing gene expression data in terms of gene sets: methodological issues, *Bioinformatics*, **23**, 980 - 987.

Gordon, A., Glazko, G., Qiu, X. and Yakovlev, A. (2007) Control of the mean number of false discoveries, Bonferroni and Stability of multiple testing., *Annual Applied Statistics*, **1**, 11.

Guillaume, J.C. (1998) [PubMed], *Ann Dermatol Venereol*, **125**, 467-468.

Harvey, B. and Levitus, M. (2009) Nucleobase-Specific Enhancement of Cy3 Fluorescence, *Journal of Fluorescence*, **19**, 443-448.

Heinlein, T., Knemeyer, J.-P., Piestert, O. and Sauer, M. (2003) Photoinduced Electron Transfer between Fluorescent Dyes and Guanosine Residues in DNA-Hairpins, *The Journal of Physical Chemistry B*, **107**, 7957-7964.

Henderson, A.R. (2006) Testing experimental data for univariate normality, *Clinica Chimica Acta*, **366**, 112-129.

Hoaglin, D.C., Mosteller, F., Tukey and J.W. (2000) Understanding Robust and Exploratory Data Analysis, *John Wiley & Sons*.

Hollander, M., Wolfe and D.A. (1999) Nonparametric Statistical Methods (second edition), *John Wiley & Sons*.

Huang, Z., Ji, D., Wang, S., Xia, A., Koberling, F., Patting, M. and Erdmann, R. (2005) Spectral Identification of Specific Photophysics of Cy5 by Means of Ensemble and Single Molecule Measurements, *The Journal of Physical Chemistry A*, **110**, 45-50.

Hubank, M. and Schatz, D.G. (1994) Identifying differences in mRNA expression by representational difference analysis of cDNA, *Nucleic Acids Research*, **22**, 5640-5648.

Hubbard, T., Andrews, D., Caccamo, M., Cameron, G., Chen, Y., Clamp, M., Clarke, L., Coates, G., Cox, T. and Cunningham, F. (2005) Ensembl 2005, *Nucleic Acids Res*, **33 Database**, D447 - 453.

Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., Cox, T., Cuff, J., Curwen, V. and Down, T. (2002) The Ensembl genome database project, *Nucleic Acids Res*, **30**, 38 - 41.

Hubbard, T.J., Aken, B.L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P. and Clarke, L. (2009) Ensembl 2009, *Nucleic Acids Res*, **37 Database**, D690 - 697.

Hubbard, T.J., Aken, B.L., Beal, K., Ballester, B., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cunningham, F. and Cutts, T. (2007) Ensembl 2007, *Nucleic Acids Res*, **35 Database**, D610 - 617.

Hunter, S., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Das, U., Daugherty, L., Duquenne, L., Finn, R.D., Gough, J., Haft, D., Hulo, N., Kahn, D., Kelly, E., Laugraud, A., Letunic, I., Lonsdale, D., Lopez, R., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Mulder, N., Natale, D., Orengo, C., Quinn, A.F., Selengut, J.D., Sigrist, C.J., Thimma, M., Thomas, P.D., Valentin, F., Wilson, D., Wu, C.H. and Yeats, C. (2009) InterPro: the integrative protein signature database, *Nucleic Acids Res*, **37**, D211-215.

Huynh, H.T., Kim, J.-J. and Won, Y. (2009) Classification Study on DNA Microarray with Feedforward Neural Network Trained by Singular Value Decomposition, *International Journal of Bio- Science and Bio- Technology*, **1**.

Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U. and Speed, T.P. (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data, *Biostatistics*, **4**, 249-264.

Ivliev, A.E., t Hoen, P.A., Villerius, M.P., den Dunnen, J.T. and Brandt, B.W. (2008) Microarray retriever: a web-based tool for searching and large scale retrieval of public microarray data, *Nucleic Acids Res*, **36 Web Server**, W327 - 331.

Jain, N., Thatte, J., Braciale, T., Ley, K., O'Connell, M. and Lee, J.K. (2003) Local-pooled-error test for identifying differentially expressed genes with a small number of replicated microarrays, *Bioinformatics*, **19**, 1945 - 1951.

Kanehisa, M. and Goto, S. (2000) KEGG: kyoto encyclopedia of genes and genomes, *Nucleic Acids Res*, **28**, 27 - 30.

Klebanov, L. and Yakovlev, A. (2007) How high is the level of technical noise in microarray data? , *Biol Direct.* , **2**.

Korn, E.L., Habermann, J.K., Upender, M.B., Ried, T. and McShane, L.M. (2004) Objective method of comparing DNA microarray image analysis systems, *Biotechniques*, **36**, 960-967.

Kumar, A. (2011) DNA CHIP TECHNOLOGY: A REVIEW, *Inter J Curr Trends Sci Tech*, **2**, 1–24.

Lai, Y. Differential expression analysis of Digital Gene Expression data: RNA-tag filtering, comparison of t-type tests and their genome-wide co-expression based adjustments, *Int J Bioinform Res Appl*, **6**, 353-365.

Leung, Y.F. and Cavalieri, D. (2003) Fundamentals of cDNA microarray data analysis, *Trends Genet*, **19**, 649-659.

Li, C. and Hung Wong, W. (2001) Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application, *Genome Biol*, **2**, RESEARCH0032.

Lin, S.M., Johnson, K.F., McConnell, P., Johnson, K. and Lockhart, D. (2002) An Introduction to DNA Microarrays. In, *Methods of Microarray Data Analysis II*. Springer US, 9-21.

Lu, J., Kerns, R.T., Peddada, S.D. and Bushel, P.R. Principal component analysis-based filtering improves detection for Affymetrix gene expression arrays, *Nucleic Acids Research*.

Maglott, D., Ostell, J., Pruitt, K.D. and Tatusova, T. (2005) Entrez Gene: gene-centered information at NCBI, *Nucleic Acids Res*, **33 Database**, D54 - 58.

Maskos, U. and Southern, E.M. (1992) Oligonucleotide hybridisations on glass supports: a novel linker for oligonucleotide synthesis and hybridisation properties of oligonucleotides synthesised in situ, *Nucleic Acids Research*, **20**, 1679-1684.

McCarthy, D.J. and Smyth, G.K. (2009) Testing significance relative to a fold-change threshold is a TREAT, *Bioinformatics*, **25**, 765-771.

Menssen, A., Edinger, G., Grun, J.R., Haase, U., Baumgrass, R., Grutzkau, A., Radbruch, A., Burmester, G.R. and Haupl, T. (2009) SiPaGene: A new repository for instant online retrieval, sharing and meta-analyses of GeneChip expression data, *BMC Genomics*, **10**, 98.

Meyer, P.E., Lafitte, F. and Bontempi, G. (2008) minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information, *BMC Bioinformatics*, **9**, 461.

Murie, C., Woody, O., Lee, A.Y. and Nadon, R. (2009) Comparison of small n statistical tests of differential expression applied to microarrays, *BMC Bioinformatics*, **10**, 45.

Mwololo, J.K., Munyua, J.K., Muturi, P.W. and Munyiri, S.W. (2010) An overview of advances in bioinformatics and its application in functional genomics, *Journal of Animal & Plant Sciences*, **6**, 645- 652.

Nadon, R. and Shoemaker, J. (2002) Statistical issues with microarrays: processing and analysis, *Trends Genet*, **18**, 265-271.

Naef, F. and Magnasco, M.O. (2003) Solving the riddle of the bright mismatches: labeling and effective binding in oligonucleotide arrays, *Phys Rev E Stat Nonlin Soft Matter Phys*, **68**, 011906.

Nagele, P. (2003) Misuse of standard error of the mean (sem) when reporting variability of a sample. A critical evaluation of four anaesthesia journals, *British Journal of Anaesthesia*, **90**, 514-516.

Nuwaysir, E.F., Huang, W., Albert, T.J., Singh, J., Nuwaysir, K., Pitas, A., Richmond, T., Gorski, T., Berg, J.P., Ballin, J., McCormick, M., Norton, J., Pollock, T., Sumwalt, T., Butcher, L., Porter, D., Molla, M., Hall, C., Blattner, F., Sussman, M.R., Wallace, R.L., Cerrina, F. and Green, R.D. (2002) Gene expression analysis using oligonucleotide arrays produced by maskless photolithography, *Genome Res*, **12**, 1749-1755.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H. and Kanehisa, M. (1999) KEGG: Kyoto Encyclopedia of Genes and Genomes, *Nucleic Acids Res*, **27**, 29 - 34.

Parkinson, H., Kapushesky, M., Shojatalab, M., Abeygunawardena, N., Coulson, R., Farne, A., Holloway, E., Kolesnykov, N., Lilja, P. and Lukk, M. (2007) ArrayExpress--a public database of microarray experiments and gene expression profiles, *Nucleic Acids Res*, **35 Database**, D747 - 750.

Parkinson, H., Sarkans, U., Shojatalab, M., Abeygunawardena, N., Contrino, S., Coulson, R., Farne, A., Lara, G.G., Holloway, E. and Kapushesky, M. (2005) ArrayExpress--a public repository for microarray gene expression data at the EBI, *Nucleic Acids Res*, **33 Database**, D553 - 555.

Pearson, R.D. (2008) A comprehensive re-analysis of the Golden Spike data: towards a benchmark for differential expression methods, *BMC Bioinformatics*, **9**, 164.

Pease, A.C., Solas, D., Sullivan, E.J., Cronin, M.T., Holmes, C.P. and Fodor, S.P. (1994) Light-generated oligonucleotide arrays for rapid DNA sequence analysis, *Proceedings of the National Academy of Sciences*, **91**, 5022-5026.

Pease, A.C., Solas, D., Sullivan, E.J., Cronin, M.T., Holmes, C.P. and Fodor, S.P. (1994) Light-generated oligonucleotide arrays for rapid DNA sequence analysis, *Proc Natl Acad Sci U S A*, **91**, 5022-5026.

Pierre, M., DeHertogh, B., DeMeulder, B., Bareke, E., Depiereux, S., Michiels, C. and Depiereux, E. (2011) Enhanced Meta-analysis Highlights Genes Involved in Metastasis from

Several Microarray Datasets, *Journal of Proteomics & Bioinformatics*, **4**, 8.

Pierre, M., DeHertogh, B., Gaigneaux, A., DeMeulder, B., Berger, F., Bareke, E., Michiels, C. and Depiereux, E. Meta-analysis of archived DNA microarrays identifies genes regulated by hypoxia and involved in a metastatic phenotype in cancer cells, *BMC Cancer*, **10**, 176.

Pierre, M., DeHertogh, B., Gaigneaux, A., DeMeulder, B., Berger, F., Bareke, E., Michiels, C. and Depiereux, E. (2010) Meta-analysis of archived DNA microarrays identifies genes regulated by hypoxia and involved in a metastatic phenotype in cancer cells, *BMC Cancer*, **10**, 176.

Pollack, J.R., Perou, C.M., Alizadeh, A.A., Eisen, M.B., Pergamenschikov, A., Williams, C.F., Jeffrey, S.S., Botstein, D. and Brown, P.O. (1999) Genome-wide analysis of DNA copy-number changes using cDNA microarrays, *Nat Genet*, **23**, 41-46.

Priness, I., Maimon, O. and Ben-Gal, I. (2007) Evaluation of gene-expression clustering via mutual information distance measure, *BMC Bioinformatics*, **8**, 111.

Rashbass, J. (1995) Online Mendelian Inheritance in Man, *Trends Genet*, **11**, 291-292.

Rayner, T.F., Rocca-Serra, P., Spellman, P.T., Causton, H.C., Farne, A., Holloway, E., Irizarry, R.A., Liu, J., Maier, D.S. and Miller, M. (2006) A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB, *BMC Bioinformatics*, **7**, 489.

Robinson, M. and Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data, *Genome Biology*, **11**, R25.

Rocca-Serra, P., Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Contrino, S., Vilo, J., Abeygunawardena, N., Mukherjee, G. and Holloway, E. (2003) Gene Expression Omnibus

ArrayExpress: a public database of gene expression data at EBI, *C R Biol*, **326**, 1075 - 1078.

Salomonis, N., Hanspers, K., Zambon, A.C., Vranizan, K., Lawlor, S.C., Dahlquist, K.D., Doniger, S.W., Stuart, J., Conklin, B.R. and Pico, A.R. (2007) GenMAPP 2: new features and resources for pathway analysis, *BMC Bioinformatics*, **8**, 217.

Santarius, T., Shipley, J., Brewer, D., Stratton, M.R. and Cooper, C.S. A census of amplified and overexpressed human cancer genes, *Nat Rev Cancer*, **10**, 59-64.

Sassolas, A., Leca-Bouvier, B.D. and Blum, L.J. (2007) DNA Biosensors and Microarrays, *Chemical Reviews*, **108**, 109-139.

Schadt, E.E., Li, C., Ellis, B. and Wong, W.H. (2001) Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data, *J Cell Biochem Suppl*, 120 - 125.

Schafer, J. and Strimmer, K. (2005) An empirical Bayes approach to inferring large-scale gene association networks, *Bioinformatics*, **21**, 754-764.

Schena, M., Shalon, D., Davis, R.W. and Brown, P.O. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray, *Science*, **270**, 467 - 470.

Schneider, W.L. and Roossinck, M.J. (2001) Genetic diversity in RNA virus quasispecies is controlled by host-virus interactions, *J Virol*, **75**, 6566-6571.

Shalon, D., Smith, S.J. and Brown, P.O. (1996) A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization, *Genome Research*, **6**, 639-645.

Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B. and Ideker, T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks, *Genome Res*, **13**, 2498-2504.

Shriner, D. and Vaughan, L.K. A unified framework for multi-locus association analysis of both common and rare variants, *BMC Genomics*, **12**, 89.

Siddiqui, A.S., Delaney, A.D., Schnerch, A., Griffith, O.L., Jones, S.J.M. and Marra, M.A. (2006) Sequence biases in large scale gene expression profiling data, *Nucleic Acids Research*, **34**, e83.

Stylianou, I.M., Affourtit, J.P., Shockley, K.R., Wilpan, R.Y., Abdi, F.A., Bhardwaj, S., Rollins, J., Churchill, G.A. and Paigen, B. (2008) Applying gene expression, proteomics and single-nucleotide polymorphism analysis for complex trait gene identification, *Genetics*, **178**, 1795-1805.

Suderman, M. and Hallett, M. (2007) Tools for visually exploring biological networks, *Bioinformatics*, **23**, 2651-2659.

Tang, T., Francois, N., Glatigny, A., Agier, N., Mucchielli, M.H., Aggerbeck, L. and Delacroix, H. (2007) Expression ratio evaluation in two-colour microarray experiments is significantly improved by correcting image misalignment, *Bioinformatics*, **23**, 2686-2691.

Torimura, M., Kurata, S., Yamada, K., Yokomaku, T., Kamagata, Y., Kanagawa, T. and Kurane, R. (2001) Fluorescence-Quenching Phenomenon by Photoinduced Electron Transfer between a Fluorescent Dye and a Nucleotide Base, *Analytical Sciences*, **17**, 155-160.

van de Wiel, M.A., Picard, F., van Wieringen, W.N. and Ylstra, B. (2010) Preprocessing and downstream analysis of microarray DNA copy number profiles, *Brief Bioinform*, **12**, 10-21.

Wang, X., Ghosh, S. and Guo, S.W. (2001) Quantitative quality control in microarray image processing and data acquisition, *Nucleic Acids Res*, **29**, E75-75.

Watson, M., Perez-Alegre, M., Baron, M.D., Delmas, C., Dovc, P., Duval, M., Foulley, J.L., Garrido-Pavon, J.J., Hulsegge, I., Jaffrezic, F., Jimenez-Marin, A., Lavric, M., Le Cao, K.A., Marot, G., Mouzaki, D., Pool, M.H., Robert-Granie, C., San Cristobal, M., Tosser-Klopp, G., Waddington, D. and de Koning, D.J. (2007) Analysis of a simulated microarray dataset: comparison of methods for data normalisation and detection of differential expression (open access publication), *Genet Sel Evol*, **39**, 669-683.

Watts, D.J. and Strogatz, S.H. (1998) Collective dynamics of 'small-world' networks, *Nature*, **393**, 440-442.

Wei, C., Li, J. and Bumgarner, R.E. (2004) Sample size for detecting differentially expressed genes in microarray experiments, *BMC Genomics*, **5**, 87.

Welford, S.M., Gregg, J., Chen, E., Garrison, D., Sorensen, P.H., Denny, C.T. and Nelson, S.F. (1998) Detection of differentially expressed genes in primary tumor tissues using representational differences analysis coupled to microarray hybridization, *Nucleic Acids Res*, **26**, 3059-3065.

Westfall, H., P. and S., Y.S. (1993) Resampling based multiple testing: Examples and methods for pvalue adjustment., *Wiley*.

Widengren, J. and Schwille, P. (2000) Characterization of Photoinduced Isomerization and Back-Isomerization of the Cyanine Dye Cy5 by Fluorescence Correlation Spectroscopy, *The Journal of Physical Chemistry A*, **104**, 6416-6428.

Wilming, L.G., Gilbert, J.G., Howe, K., Trevanion, S., Hubbard, T. and Harrow, J.L. (2008) The vertebrate genome annotation (Vega) database, *Nucleic Acids Res*, **36 Database**, D753 - 760.

Wilson, R., Cossins, A.R. and Spiller, D.G. (2006) Encoded Microcarriers For High-Throughput Multiplexed Detection, *Angewandte Chemie International Edition*, **45**, 6104-6117.

Workman, C., Jensen, L.J., Jarmer, H., Berka, R., Gautier, L., Nielser, H.B., Saxild, H.H., Nielsen, C., Brunak, S. and Knudsen, S. (2002) A new non-linear normalization method for reducing variability in DNA microarray experiments, *Genome Biol*, **3**, research0048.

Wouters, L., Gohlmann, H.W., Bijnens, L., Kass, S.U., Molenberghs, G. and Lewi, P.J. (2003) Graphical exploration of gene expression data: a comparative study of three multivariate methods, *Biometrics*, **59**, 1131-1139.

Wu, X., Dong, H., Luo, L., Zhu, Y., Peng, G., Reveille, J.D. and Xiong, M. A novel statistic for genome-wide interaction analysis, *PLoS Genet*, **6**.

Wu, Z., Irizarry, R., Gentleman, R., Murillo, F. and Spencer, F. (2004) A model-based background adjustment for oligonucleotide expression arrays, *Journal of the American Statistical Association*, **99**, 909 - 917.

Wu, Z., Irizarry, R.A., Gentleman, R., Martinez-Murillo, F. and Spencer, F. (2004) A model-based background adjustement for oligonucleotides expression arrays, *Journal of the American Statistical Association*, **99**, 909-917.

XiaoKun, T. and HuaSheng, X. (2009) Perspectives of DNA microarray and next-generation DNA sequencing technologies, *Science in China Series C: Life Sciences*, **52**, 7-16.

Yamasaki, C., Murakami, K., Takeda, J., Sato, Y., Noda, A., Sakate, R., Habara, T., Nakaoka, H., Todokoro, F. and Matsuya, A. (2009) H-InvDB in 2009: extended database and data mining resources for human genes and transcripts, *Nucleic Acids Res*, **38 Database**, D626 - 632.

Yang, I.V., Chen, E., Hasseman, J.P., Liang, W., Frank, B.C., Wang, S., Sharov, V., Saeed, A.I., White, J. and Li, J. (2002) Within the fold: assessing differential expression measures and reproducibility in microarray assays, *Genome Biol*, **3**, research0062.

Yarden, Y. and Sliwkowski, M.X. (2001) Untangling the ErbB signalling network, *Nat Rev Mol Cell Biol*, **2**, 127 - 137.

Yin, J., McLoughlin, S., Jeffery, I.B., Glaviano, A., Kennedy, B. and Higgins, D.G. Integrating multiple genome annotation databases improves the interpretation of microarray gene expression data, *BMC Genomics*, **11**, 50.

Young, I.T. (1977) Proof without prejudice: use of the Kolmogorov-Smirnov test for the analysis of histograms from flow systems and other sources, *Journal of Histochemistry & Cytochemistry*, **25**, 935-941.

Yu, H., Luscombe, N.M., Qian, J. and Gerstein, M. (2003) Genomic analysis of gene expression relationships in transcriptional regulatory networks, *Trends Genet*, **19**, 422-427.

Zhu, Y., Zhu, Y. and Xu, W. (2008) EzArray: A web-based highly automated Affymetrix expression array data management and analysis system, *BMC Bioinformatics*, **9**, 46.

Appendices

List of acronyms

ADF	Array Design Format
AE	Array Express
AJAX	Asynchronous JavaScript and XML
ANOVA	ANalysis Of VAriance
API	Application Programing Interface
BMC	Biomedical Center
CDM	Conceptual Diagram Model
cDNA	complementary Desoxyribonucleic Acid
CGI	Common Gateway interface
CL	Cyclic Loess
cRNA	complementary Ribonucleic Acid
CSS	Cascading Style Sheet
DAVID	Database for Annotation, Visualization and Integrated Discovery
DBMS	Database Management Systems
DNA	Desoxyribonucleic Acid
EBI	European Bioinformatics Institute
EER	Enhaced Entity Relation
ER	Entity Relation
FDR	False Discovery Rate
FPR	False Positive Rate
FWER	Family-Wise Error Rate
GCRMA	GC Robust Multi-array Average
GEO	Gene Expression Omnibus
GO	Gene Ontology
HTTP	Hypertext Transfer Protocol
IM	Ideal Mismatch
JUNG	Java Universal Network Graph
KEGG	Kyoto Encyclopedia of Genes and Genoms
LAMP	Linux Apache MySQL PHP/Pythom/Perl
LDC	Logical Diagram Model
LPE	Local Pooled Error
MAGE	Microarray Gene Expression
MaRe	Microarrary Retriever
MAS	Microarray Affymetrix Suite
MM	Mismatch
mRNA	messenger Ribonucleic Acid
MVC	Model-View-Controller
NCBI	National Center of Biotechnology
OMIM	Online Mendelian Inheritance in Man
OOP	Object Oriented Programming
OS	Operating System
PDM	Physical Diagram Model
PM	Perfect Match
RMA	Robust Multichip Areraging
RHEL	Red Hat Enterprise Linux
RNA	Ribonucleic Acid
rRNA	ribosomal Ribonucleic Acid
SB	Specific Background
SD	Standard Deviation
SDRF	Sample and Data Relationship Format
SOFT	Simple Ombibus Format on Text
tRNA	transfer Ribonucleic Acid
VEGA	Vebrate Genome Annotation

List of tables

List of figures

List of equations

List of publications

Publications presented in this thesis

1. **PathEx: A novel multi factors based datasets selector web tool**. *BMC Bioinformatics* 2010, 11:528doi:10.1186/1471-2105-11-528
   Eric BAREKE, Michael PIERRE, Bertrand DE MEULDER, Anthoula GAIGNEAUX, Sophie DEPIEREUX, Naji Habra, Eric DEPIEREUX
2. **gViz, a novel co-expression networks visualization tool** (Submitted to BMC Research Notes: *Under Review*), 2011
   Raphaël Helaers*, **Eric Bareke***, Bertrand De Meulder, Michael Pierre, Sophie Depiereux, Naji Habra and Eric Depiereux

Other publications with author contribution

1. **Enanced Meta-analysis Highlights Genes Involved in Metastasis from Several Microarray Datasets**. *Journal of Proteomics and Bioinformatics*, 2011 DOI: 10.4172/jpb.1000164 Pierre M, DeHertogh B, DeMeulder B, **Bareke E**, Depiereux S, Michiels C, Depiereux E.
2. **Functional Analysis: Evaluation of Response Intensities - Tailoring ANOVA for Lists of Expression Subsets**.FAERI-TALES. *BMC Bioinformatics* 2010, 11:510doi:10.1186/1471-2105-11-510 Fabrice BERGER, Bertrand DE MEULDER, Anthoula GAIGNEAUX, Sophie DEPIEREUX, **Eric BAREKE**, Michael PIERRE, Benoit DE HERTOGH, Mauro DELORENZI, Eric DEPIEREUX
3. **A benchmark for statistical microarray data analysis that preserves actual biological and technical variance**. *BMC Bioinformatics* 2010, 11:17doi:10.1186/1471-2105-11-17 Benoit DE HERTOGH, Bertrand DE MEULDER, Fabrice BERGER, Michael PIERRE, **Eric BAREKE**, Anthoula GAIGNEAUX, Eric DEPIEREUX
4. **Meta-analysis of archived DNA microarrays identifies genes regulated by hypoxia and involved in a metastatic phenotype in cancer cells**. *BMC Cancer* 2010, 10:176doi:10.1186/1471-2407-10-176 Michael PIERRE, Benoit DE HERTOGH, Anthoula GAIGNEAUX, Bertrand DE MEULDER, Fabrice BERGER, **Eric BAREKE**, Carine MICHIELS, Eric DEPIEREUX
5. **PHOENIX, a web interface for (re)analysis of microarray data**. *Central European Journal of Biology*, volume 4, issue 4, pp. 603-618 Fabrice BERGER, Benoit DE HERTOGH, **Eric BAREKE**, Michael PIERRE, Anthoula GAIGNEAUX, Eric DEPIEREUX

Supplementary tables

Table 6 - PathEx JavaScript methods

Pathex JavaScript Methods	
Methods	**Description**
get_grid()	Return the object of the grid that contains the element.
get_row()	Return the object of the row that contains the element.
get_tableview()	Return the object of the tableview that contain the element.
grid_cancel_edit()	Cancel editing for the row that contains the element.
grid_cancel_insert()	Cancel inserting new row for the tableview that contains the element.
grid_collapse()	Make the row that contains the element collapse its table details.
grid_confirm_edit()	Confirm editing for the row that contains the element.
grid_confirm_insert()	Confirm inserting new row for the tableview that contains the element.
grid_delete()	Delete the row that contains the element.
grid_edit()	Start editing the row which contains the element.
grid_expand()	Make the row that contains the element expand its table details.

grid_gopage()	**Make tableview that contain the element navigate to certain page.**
grid_insert()	**Start inserting new row for the tableview that contains the element.**
grid_refresh()	**Make the grid that contains the element refresh.**
grid_sort()	**Sort the column with specified id.**
grid_toggle_select()	**Toggle selection for the row that contains the element.**

Table 7 - PathEx data dictionary

Name	Description	Code	Data Type	Length
chromosome	Unique chromosome accession ID	chromosome	Variable characters (5)	5
g2p_id	Unique gene accession ID associated with a Platform	g2p_id	Integer	
gene_name	Unique name describing a Gene	gene_name	Variable characters (200)	200
gene_symbol	NCBI EntrezGene assigned Gene Symbol	gene_symbol	Variable characters (15)	15
omim_number	NCBI OMIM assigned number	omim_number	Variable characters (20)	20
organism_id	Unique organism accession ID	organism_id	Integer	

organism_name	Scientific name associated to an organism	organism_name	Variable characters (200)	200
p2s_id	Unique Platform accession number associated with a Sample	p2s_id	Integer	
path_id	Pathway database automatically assigned internal ID	path_id	Variable characters (20)	20
path_name	Unique name describing Pathway	path_name	Variable characters (200)	200
pid	Platform database automatically assigned internal ID	pid	Variable characters (20)	20
pl	Unique Platform accession number approved and issued by GEO, AE	pl	Variable characters (10)	10
pl_distribution	Microarrays are 'commercial', 'non-commercial', or 'custom-commercial' in accordance with how the array was	pl_distribution	Text	

	manufactured.			
pl_manufacturer	Name of the company, facility or laboratory where the array was manufactured or produced	pl_manufacturer	Text	
pl_sub_date	Platform date submitted	pl_sub_date	Date	
pl_technology	the category describing the Platform technology: spotted DNA/cDNA, spotted oligonucleotide, in situ oligonucleotide, antibody, tissue, SARST, RT-PCR, MS, or MPSS	pl_technology	Text	
pl_title	Platform unique name describing the Platform	pl_title	Text	
pl_upd_date	Platform date last updated	pl_upd_date	Date	
pubmed_id	NCBI PubMed identifier (PMID)	pubmed_id	Variable characters (20)	20
px_id	Series database automatically assigned internal ID	px_id	Integer	
s2s_id	unique Series accession number associated with a Sample	s2s_id	Integer	
se	Series unique accession number approved and issued by GEO,	se	Variable	10

	AE		characters (10)	
se_status	Series date released to public	se_status	Text	
se_sub_date	Series date submitted	se_sub_date	Date	
se_summary	A description of the goals and objectives of a study	se_summary	Text	
se_title	Unique name describing the overall study	se_title	Text	
se_type	Keyword(s)generally describing the type of study, e.g., time course, dose response, comparative genomic hybridization, ChIP-chip, cell type comparison, disease state analysis, stress response, genetic modification, etc.	se_type	Variable characters (45)	45
se_upd_date	Series date last updated	se_upd_date	Date	
sec_id	Sample database automatically assigned internal ID	sec_id	Variable characters (32)	32
sm	unique Sample accession number approved and issued by GEO, AE	sm	Variable characters	10

			(10)	
sm_count_channel	number of labeling channels, could be 1 or 2	sm_count_channel	Integer	
sm_description	List of characteristics of the biological source, including factors not necessarily under investigation, e.g., Strain: C57BL/6, Gender: female, Age: 45 days, Tissue: bladder tumor, Tumor stage: Ta. Multiple characteristics columns can be included	sm_description	Text	
sm_row_count	Number of data rows	sm_row_count	Integer	
sm_source1	Name to identify the biological material and the experimental variable(s), e.g., vastus lateralis muscle, exercised, 60 min	sm_source1	Text	
sm_source2	Name to identify the biological material and the experimental variable(s), e.g., vastus lateralis muscle, exercised, 60 min	sm_source2	Text	
sm_status	Sample date released to public	sm_status	Text	
sm_sub_date	Sample date submitted	sm_sub_date	Date	
sm_title	Unique name describing a Sample	sm_title	Text	
sm_type	Type of samples, values in current database are genomic, mixed,	sm_type	Text	

	MPSS, protein, RNA, SAGE, SARST, other			
sm_upd_date	Sample date last updated	sm_upd_date	Date	
source_code	Data source automatically assigned internal ID	source_code	Variable characters (45)	45
source_data_type	Type of data source: gene, pathway,…	source_data_type	Variable characters (45)	45
source_name	Data source name	source_name	Variable characters (100)	100

Table 8 -List of datasets drawn by PathEx in the course of case study validation (Meta-analysis)

Experiment/Study Accession numbers (keywords used)	Platform	Source	Availability	Experimental conditions
E-GEOD-1323 (metastasis)	HG-U133A	AE	Available	3 human colorectal cancer derived from a primary tumor VS. 3 corresponding lymph node metastases
E-GEOD-2280 (metastasis)	HG-U133A	AE	Available	8 squamous cell carcinoma of the oral cavity VS. 19 corresponding lymph node metastases
E-MEXP-44	HG-U95Av2	AE	Available	15 head and neck squamous cell carcinoma VS. 3 corresponding lymph node metastases
	HG-UgeneFL			12 head and neck squamous cell carcinoma VS. 11 corresponding lymph node metastases
GSE1056	HG-U95Av2	GEO	Not available for public use	2 human hepatocellular carcinoma under hypoxia for 2 hours VS. 2 control human hepatocellular carcinoma
				2 human hepatocellular carcinoma under hypoxia for 24 hours VS. 2 control human hepatocellular carcinoma
GSE2280 (metastasis)	HG-U133A	GEO	Available	22 squamous cell carcinoma of the oral cavity VS. 5 corresponding lymph node metastases

GSE2603 (Not maching provided keywords!!!)	**HG-U133A**	**GEO**	**Available**	**100 primary breast cancer VS. 21 lung metastases**
GSE3325 (metastasis)	**HG-U133Plus2.0**	**GEO**	**Available**	**7 primary prostate cancer VS. 6 metastases**
GSE4086 (hypoxia)	**HG-U133Plus2.0**	**GEO**	**Available**	**2 human Burkitt's lymphoma under hypoxia VS. 2 control human Burkitt's lymphoma**
GSE468	**HC-G110**	**GEO**	**Available**	**13 primary medulloblastomas VS. 10 metastatic medulloblastomas**
GSE4840	**HG-U133A**	**GEO**	**Not available for public use**	**3 samples from normal melanocyte culture VS. 12 samples from culture of cutaneous metastasis of melanoma**
	HG-U133B			**3 samples from normal melanocyte culture VS. 12 samples from culture of cutaneous metastasis of melanoma**
GSE4843	**HG-U133Plus2.0**	**GEO**	**Not available for public use**	**45 samples from culture of cutaneous melanoma metastasis**
GSE6369 (prostate)	**HG-U133Plus2.0**	**GEO**	**Available**	**1 primary prostate carcinoma VS. 1 metastatic prostate carcinoma**
GSE6919 (prostate)	**HG-U95Av2**	**GEO**	**Available**	**65 primary prostate tumors VS. 25 metastatic prostate**

				tumors
	HG-U95B			**66 primary prostate tumors VS. 25 metastatic prostate tumors**
	HG-U95C			**65 primary prostate tumors VS. 25 metastatic prostate tumors**
GSE7929 (metastasis)	**HG-U133A**	**GEO**	**Available**	**11 poorly metastatic melanoma VS. 21 highly metastatic melanoma**
GSE7930 (metastasis)	**HG-U133A**	**GEO**	**Available**	**3 poorly metastatic prostate tumors VS. 3 highly metastatic prostate tumors**
GSE7956 (metastasis)	**HG-U133A**	**GEO**	**Available**	**10 poorly metastatic melanoma VS. 29 highly metastatic melanoma**
GSE8401 (metastasis)	**HG-U133A**	**GEO**	**Available**	**31 primary melanoma VS. 52 melanoma metastasis**

Table 9 - List of metadata sets drawn by PathEx in the course of proofing stage validation (Enhanced meta-analysis study)

Meta-dataset Name	Experimental conditions	GeneChip models	Datasets
Meta-dataset 1	Primary tumor, normal tissue, poorly metastatic tissue VS. metastasis, highly metastatic tissue	HG-U133A	E-GEOD-1323, E-GEOD-2280, GSE2280, GSE2603, GSE4840 (HG-U133A), GSE7929, GSE7930, GSE7956, GSE8401
Meta-dataset 2	Primary tumor, poorly metastatic tissue VS. metastasis, highly metastatic tissue	HG-U133A	E-GEOD-1323, E-GEOD-2280, GSE2280, GSE2603, GSE7929, GSE7930, GSE7956, GSE8401
Meta-dataset 3	Primary tumor, normal tissue VS. metastasis	HG-U133A	E-GEOD-1323, E-GEOD-2280, GSE2280, GSE2603, GSE4840 (HG-U133A), GSE7929, GSE7956, GSE8401
Meta-dataset 4	Primary tumor VS. metastasis	HG-U133A	E-GEOD-1323, E-GEOD-2280, GSE2280, GSE2603, GSE4840 (HG-U133A), GSE7929, GSE7956, GSE8401

Meta-dataset 5	**Primary tumor VS. metastasis**	**HG-U133A**	**E-GEOD-1323, E-GEOD-2280, GSE2280, GSE2603, GSE7929, GSE7956, GSE8401**
Meta-dataset 6	**Squamous cell carcinoma of the oral cavity VS. corresponding lymph node metastases**	**HG-U133A**	**E-GEOD-2280, GSE2280**
Meta-dataset 7	**Normal melanocyte culture, poorly metastatic melanoma, primary melanoma VS. culture of cutaneous metastasis of melanoma, highly metastatic melanoma, melanoma metastasis**	**HG-U133A**	**GSE4840 (HG-U133A), GSE7929, GSE7956, GSE8401**
Meta-dataset 8	**Poorly metastatic melanoma, primary melanoma VS. culture of cutaneous metastasis of melanoma, highly metastatic melanoma, melanoma metastasis**	**HG-U133A**	**GSE4840 (HG-U133A), GSE7929, GSE7956, GSE8401**
Meta-dataset 9	**Poorly metastatic melanoma, primary melanoma VS. highly metastatic melanoma, melanoma metastasis**	**HG-U133A**	**GSE7929, GSE7956, GSE8401**
Meta-dataset 10	**Primary tumor VS. metastasis**	**HG-U95Av2**	**E-MEXP-44 (HG-U95Av2), GSE6919 (HG-U95Av2)**
Meta-dataset 11	**Hypoxia VS. normoxia**	**HG-U95Av2**	**GSE1056**

Meta-dataset 12	**Primary tumor, normoxia VS. metastasis, hypoxia**	**HG-U133Plus2.0**	**GSE3325, GSE4086, GSE4843, GSE6369**
Meta-dataset 13	**Primary tumor VS. metastasis**	**HG-U133Plus2.0**	**GSE3325, GSE4843, GSE6369**
Meta-dataset 14	**Primary prostate cancer VS. metastases**	**HG-U133Plus2.0**	**GSE3325, GSE6369**

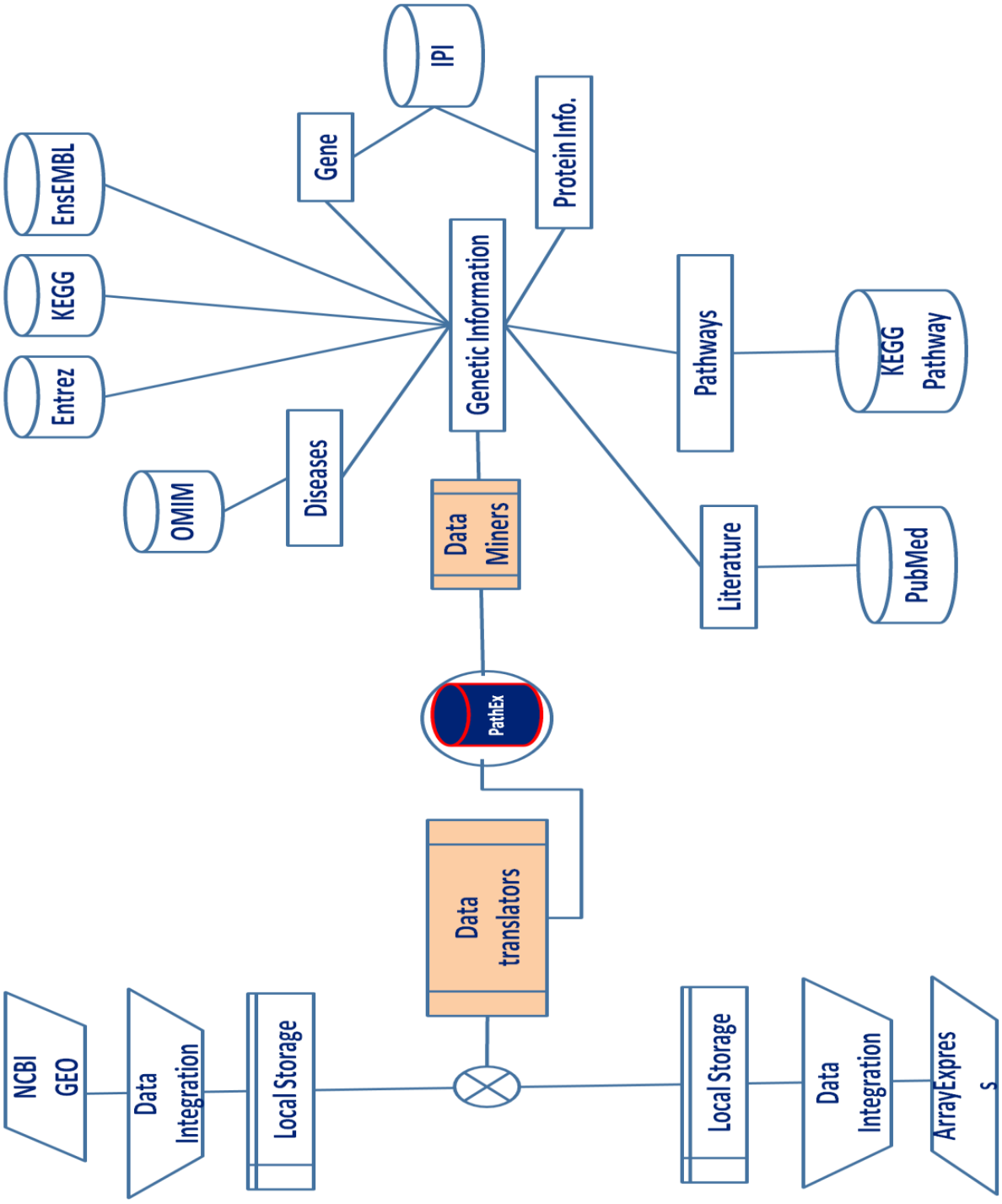
PathEx data sources

Table 10 - Sources and status of information integrated into PathEx

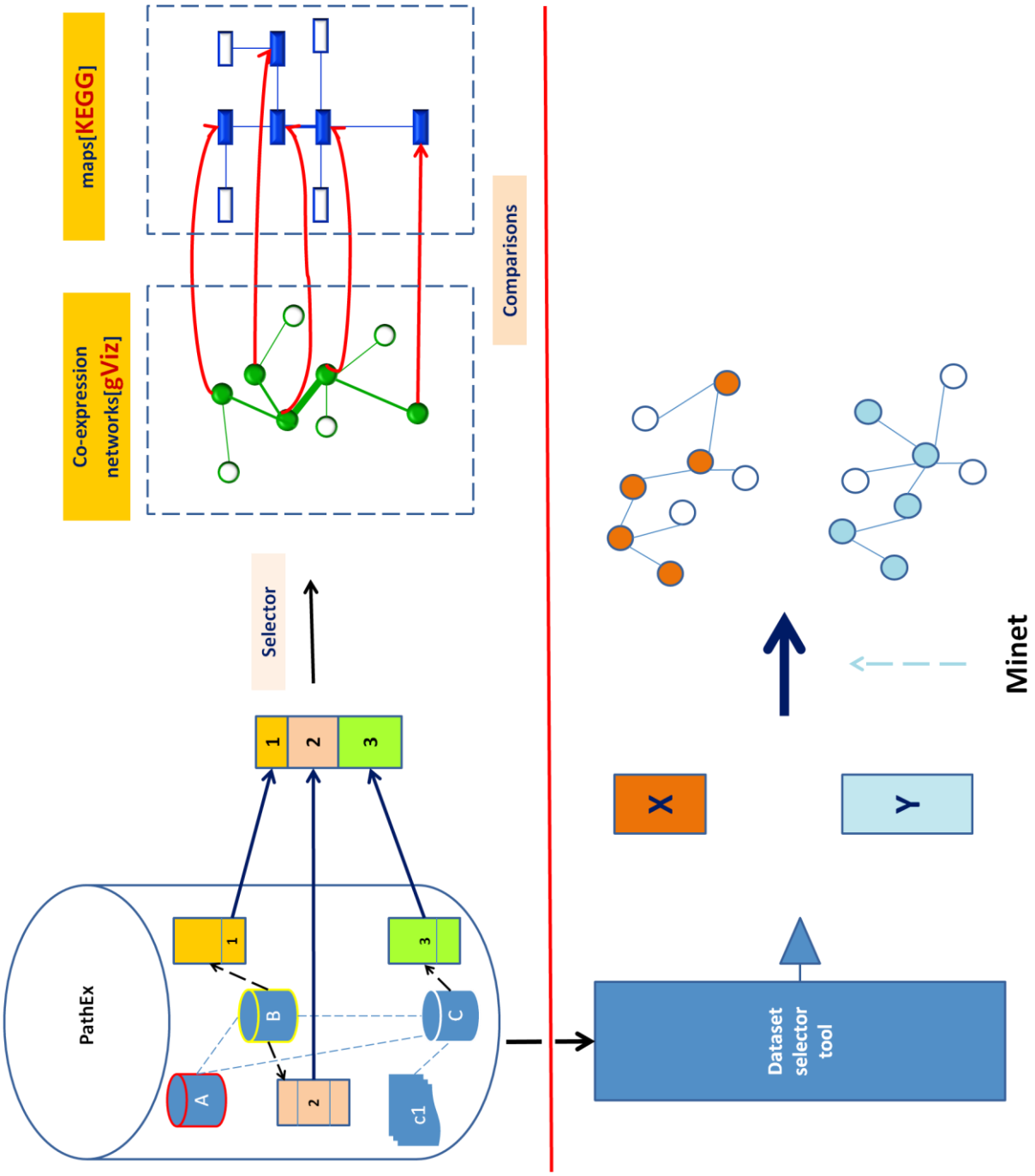
No.	Source Name	Data Type	PathEx Last Update	Integrated
1	Entrez Gene	gene	December 14, 2010	Yes
2	ENSEMBL	gene	December 14, 2010	Yes
3	EMBL	protein_xref	December 14, 2010	Yes
4	KEGG Gene	gene	December 14, 2010	Yes
5	UniGene	gene_xref	December 14, 2010	Yes
6	UniProtKB/Swiss-Prot	protein	December 14, 2010	Yes
7	UniProtKB/TrEMBL	protein	December 14, 2010	Yes
8	ENSEMBL	protein	December 14, 2010	Yes
9	Entrez Gene	interaction	December 14, 2010	Yes
10	OMIM	interaction	December 14, 2010	Yes
11	Genome Expression Omnibus	array expression	December 14, 2010	Yes
12	Array Express	array expression	December 14, 2010	Yes
13	PubMed	citations	December 14, 2010	Yes
14	Gene Ontology	ontology	December 14, 2010	Yes
15	GenMAPP	pathways	December 14, 2010	Perspective
16	BioCarta	pathways	December 14, 2010	In process
17	WikiPathways	pathways	December 14, 2010	In process
18	KEGG Pathway	pathway	December 14, 2010	Yes

Additional illustrations

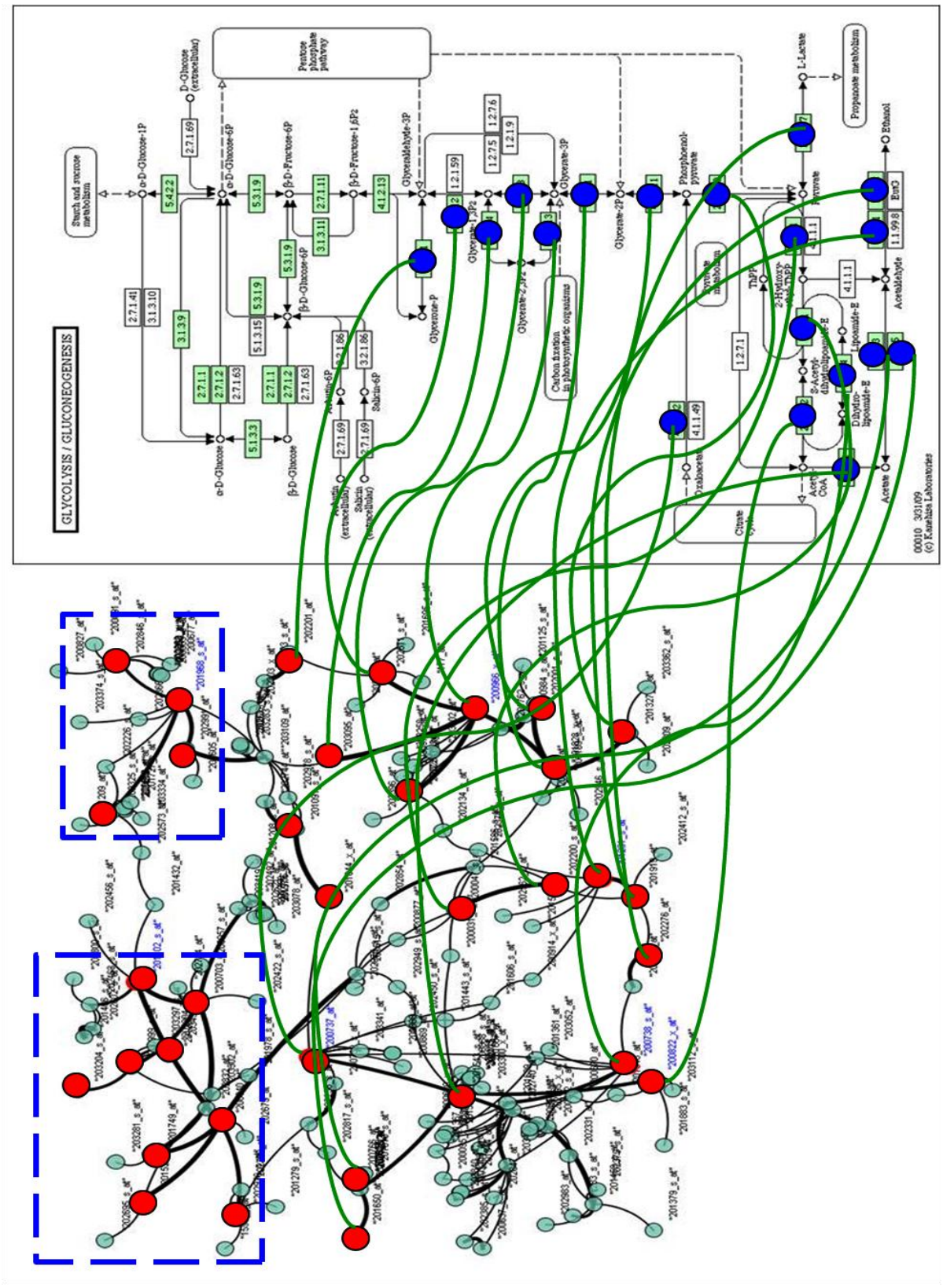
PathEx data integration approach rationale



PathEx future dataset processing approach



PathEx future direction expected outcome maps example



PathEx web tool availability & deployment procedures

PathEx core class deployment

1. Unzip **PathExCoreSuite.zip** file from the CD.
2. Copy **PathExCoreSuite** folder inside unzipped directory to your web server folder.
3. Make sure that you have installed MySQL and **PHP5**.
4. Start browsing the suite in web browser with url **"http://your-web-server-folder/ PathExCoreSuite /index.php"**.

PathEx database component deployment

1. Unzip **PathExDB.zip** file from the CD.
2. Make sure that you have installed MySQL GUI Tools installed to your computer.
3. Restore the unzipped file through MySQL GUI Administrator Tool

PathEx interfaces deployment

1. Unzip **pathex.zip** file from the CD.
2. Move unzipped (**pathex.zip)** directory to your web server folder.
3. Start browsing the suite in web browser with url **"http://your-web-server-folder/ pathex/index.php"**.

Dataset builder deployment

1. Unzip **PathexBuilder.zip** file from the CD.
2. Move unzipped directory out of the web server folder.
3. Edit the "**config.tab**" file appropriately

Note: You will require contacting us to guide you how to import samples files used to build datasets. Make sure you have enough space on your server: "**at least**" **4TB disk space**

gViz availability & deployment procedures

Availability

gViz is available at http://urbm-cluster.urbm.fundp.ac.be/webapps/gviz for 32 and 64 bit Windows, MacOS X and Linux/Unix.

Deployment procedures

It requires Java engine 1.6 or higher to run (http://java.com).

To connect with the external database PathEx, the port 3306 of the user's computer must be opened. This port may be blocked by a firewall and thus users must ask their network administrator to unblock it for their machine, if necessary. gViz is available under the Open GPL license. Install instructions can be downloaded from either inside the application itself or from the web page of the application above mentioned.